Предисловие

В курсе «Исследование операций» (ИСО) рассматриваются задачи принятия решений в организационных системах. Исследование операций интенсивно развивалось в годы второй мировой войны, а также в послевоенные годы и получило признание как самостоятельный раздел науки к середине 50-х годов XX века. В настоящее время ИСО широко применяется во всех сферах человеческой деятельности, поскольку в сложных организационных системах невозможно принятие оптимальных решений без математического моделирования и использования компьютеров.

Имеется большое количество прекрасных учебников, посвященных ИСО. Однако, к сожалению, до настоящего времени нет ни одного учебника, который охватывал бы все вопросы, входящие в программу курса ИСО. В некоторых учебниках отдельные разделы ИСО изложены слишком подробно, требуется освоить такой большой объем информации, что студенту порой трудно отделить главное от второстепенного. Есть учебники, посвященные специальным классам задач ИСО, в них рассматриваются узко специфические вопросы, требующие углубленного изучения с применением сложного математического аппарата. Эти учебники могут быть весьма полезными для спецкурсов, но не пригодны для использования в общем курсе ИСО. Вот почему возникла необходимость написания учебника по курсу ИСО, где был бы собран и компактно изложен обширный материал, разбросанный по многочисленным учебникам. Материал лекций излагается достаточно кратко, чтобы дать студентам общее представление о проблемах, возникающих при рассмотрении различных классов задач ИСО. Список литературы, использованной в процессе разработки данного лекционного курса по ИСО, приведен в конце учебника.

Учебное пособие разработан в соответствии с государственными стандартами и типовыми программами. Представленный в нем лекционный материал охватывает различные разделы иследования операции. Некоторые вопросы могут быть вынесены для самостоятельной работы студентов.

1. ВВЕДЕНИЕ В ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Линейное программирование – это раздел математики и изучает методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменными.

Термин «программирование» в названии дисциплины ничего общего с термином «программирование (т.е. составление программ) для ЭВМ» не имеет, так как дисциплина «линейное программирование» возникла еще до того времени, когда ЭВМ стали широко применяться при решении математических, инженерных, экономических и других задач. Термин «линейное программирование» возник в результате неточного перевода английского «linear programming». Одно из значений слова «programming» – составление планов, планирование. Следовательно, правильным переводом «linear programming» было бы не «линейное программирование», а «линейное планирование», что более точно отражает содержание дисциплины.

Линейное программирование возникло после второй мировой войны и стало быстро развиваться, привлекая внимание математиков, экономистов и инженеров благодаря возможности широкого практического применения, а также математической стройности.

Можно сказать, что линейное программирование применимо для решения математических моделей тех процессов и систем, в основу которых может быть положена гипотеза линейного представления реального мира.

1.1. Математическая модель задач линейного программирования

Математической моделью задачи называется совокупность математических соотношений, описывающих суть задачи. Составление математической модели включает:

- выбор переменных задачи;
- составление системы ограничений;
- выбор целевой функции.

Рассмотрим формулировки задачи использования ресурсов или планирования производства: для изготовления n видов продукции $P_1, P_2, ..., P_n$ используется m видов сырья (ресурсов) $S_1, S_2, ..., S_m$. Расход ресурсов на единицу каждого вида продукции известен a_{ij} $(i = \overline{1,m}; j = \overline{1,n})$.

Также известна прибыль от реализации продукции каждого вида $C_1, C_2, ..., C_n$. Требуется составить план выпуска продукции обеспечивающий максимальную выгоду.

Составим математическую модель этой задачи:

1. Выбираем переменные задачи: пусть $x_1, x_2, ..., x_n$ – это количество продукции каждого вида, причем $x_{ij} \geq 0$.
2. Заносим исходные данные в таблицу:

Ресурсы	Расход ресурсов на производство единицы продукции				Запас ресурсов
	P_1	P_2	…	P_n	
S_1	a_{11}	a_{12}	…	a_{1n}	b_1
S_2	a_{21}	a_{22}	…	a_{2n}	b_2
…	…	…	…	…	…
S_m	a_{m1}	a_{m2}	…	a_{mn}	b_m
Прибыль	c_1	c_2	…	c_n	max

3. Составляем систему ограничений задачи: ограничения задачи связаны с ресурсами имеющимся в наличии, поэтому система будет содержать m-ограничений по каждому ресурсу.

$$S_1 : a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1$$

так как мы не можем израсходовать ресурсов больше, чем имеется в наличии ставится знак “≤”, следовательно, аналогично и для остальных ресурсов.

$$S_2 : a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leq b_2$$

....

$$S_m : a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \leq b_m$$

4. Целевая функция.Так как цель задачи максимальная прибыль от реализации всей продукции, то

$$Z(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \to \max$$

Таким образом, математическая модель этой задачи имеет вид: найти такой план $X = (x_1, x_2, ..., x_n)$ выпуска продукции удовлетворяющий системе ограничений :

$$\begin{cases} a_{11}x_1 + a_{21}x_2 + ... + a_{1n}x_n \le b_1 \\ ----------------- \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \le b_m \end{cases}$$

и условию неотрицательности $x_j \ge 0$, при котором целевая функция

$$Z(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \to \max.$$

1.2. Стандартная и каноническая форма задач линейного программирования

Чаще всего при составлении математической модели задачи получается система ограничений из неравенств и не на все переменные может накладываться условие не отрицательности. В этом случае задача имеет *стандартную форму*. Для решения задачи аналитическим методом требуется, чтобы в системе были только уравнения и все переменные были не отрицательными.

Определение 1.1. *Если все ограничения задачи являются уравнениями и все переменные удовлетворяют условию не отрицательности, то задача записана в **канонической форме**.*

Таким образом, каноническая форма имеет вид:

$$Z(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \to extr$$

$$\begin{cases} a_{11}x_1 + a_{21}x_2 + ... + a_{1n}x_n = b_1 \\ ----------------- \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m \end{cases}$$

$$x_j \le 0, \quad j = \overline{1,n}$$

Если система содержит неравенства, то от неравенства к уравнению переходят следующим образом:

Вводят дополнительные переменные в левые части неравенства;

- если знак неравенства “≤”, то переменная берется с коэффициентом “+1”;
- если знак “≥”, то дополнительная переменная “-1”.

В целевую функцию эти переменные записываются с нулевыми коэффициентами.

Если на какую либо переменную не наложено условие не отрицательности, то ее заменяют на разность двух других переменных, каждое из которых неотрицательное.

1.3. Алгоритм симплексного метода

1. Математическая модель задачи должна иметь каноническую форму. Если она записана в стандартной форме, то ее нужно привести к канонической.

2. Находим базисные вектора, т.е. начальное опорное решение. Количество базисных векторов совпадает с количеством ограничении и при расположении по порядку выбора базисные вектора представляет единичную матрицу. Для проверки оптимальности решения заполняем ***симплексную таблицу***.

Ба	$Z_б$	C_0	C_1	C_2	…	C_n	C_{m+1}	…	C_{n+m}
		A_0	A_1	A_2	…	A_n	A_{n+1}	…	A_{n+m}
					…			…	
					…			…	
					…			…	
					…			…	
	Δк				…			…	

Все строки таблицы 1-го шага, за исключением строки Δ*к* (индексной строки), заполняют по данным системы ограничений и целевой функции.

Проверяем начальное опорное решение на оптимальность. Индексная строка находится по формуле $\Delta_к = \sum C_б A_j - c_j$. При решении задачи возможны следующие случаи:

1) при решении задачи на *минимум:*

а) все оценки $\Delta_к \leq 0$, тогда найденное решение оптимальное;

б) хотя бы одна оценка Δк >0, но при соответствующей переменной нет ни одного положительного коэффициента, тогда решение задачи прекращаем, т.к. $\min Z(X) = -\infty$, т.е. целевая функция не ограничена в области допустимых решений;

в) хотя бы одна оценка Δк > 0 и при соответствующей переменной есть хотя бы один положительный коэффициент, тогда можно перейти к другому опорному решению, которому соответствует меньшее значение целевой функции.

2) при решении задачи на *максимум* можно поменять все знаки целевой функции на противоположные и решать задачу на минимум, при записи ответа обратно поменять все знаки.

3) В случае необходимости строим новое опорное решение. Для этого необходимо найти ключевой столбец, ключевую строку и ключевой элемент. Ключевой столбец указывает на переменную, которую следует

ввести в число базисных переменных следующего опорного решения. Ключевая строка указывает на переменную, которую следует вывести из базисных переменных для улучшения решения. Ключевой элемент нужен для расчета элементов новой симплексной таблицы, соответствующей улучшенному опорному решению. Их нахождение зависит от цели задачи.

При решении задачи на *минимум:*

а) *ключевым столбцом* является столбец соответствующий наибольшей отрицательной оценке $A_к$ в индексной строке;

б) *ключевой строкой* является строка соответствующая минимальному отношению свободных членов к положительным коэффициентам ключевого столбца, т.е.

$$\min\theta = \frac{A_0}{\{\textit{положит.эл.кл.столбца}\}}$$

в) *ключевым элементом* является число, расположенное на пересечении ключегого столбца и ключевой строки;

4) Заполняем следующую симплексную таблицу.

а) переписываем ключевую строку, разделив ее на ключевой элемент;

б) заполняем базисные столбцы: вместо ключевого элемента запишем единицу, остальные элементы зануляются;

в) остальные коэффициенты таблицы находим по правилу «прямоугольника».

Правило «прямоугольника» заключается в следующем:

$$\textit{нов.эл.} = \textit{стар эл.} - \frac{\textit{соотв.эл.кл.столб} \times \textit{соотв.эл.кл.стр}}{\textit{ключевой элемент}}$$

Получаем новое опорное решение.

5) Если решление не оптимально возвращаемся к третьему шагу алгоритма.

1.3.1. Решение задачи симплексным методом

А) Решить задачу линейного программирования

$$z(x) = 6x_1 - 3x_2 + 5x_3 - 2x_4 - 4x_5 + 2x_6 \to \min$$
$$x_1 + 2x_2 + x_4 + 3x_5 = 17$$
$$4x_2 + x_3 + x_5 = 12$$
$$x_2 + 8x_4 - x_5 + x \ = 6$$
$$x_j \geq 0,\ j = \overline{1,6}$$

Матрица A, векторы b и c равны

$$A = \begin{pmatrix} 1 & 2 & 0 & 1 & 3 & 0 \\ 0 & 4 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 8 & -1 & 1 \end{pmatrix} = \left(a^1, a^2, a^3, a^4, a^5, a^6\right),$$

$$a^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \; a^2 = \begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix}, \; a^3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \; a^4 = \begin{pmatrix} 1 \\ 0 \\ 8 \end{pmatrix},$$

$$a^5 = \begin{pmatrix} 3 \\ 1 \\ -1 \end{pmatrix}, \; a^6 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \; b = \begin{pmatrix} 17 \\ 12 \\ 6 \end{pmatrix}, \quad c' = (6, -3, 5, -2, -4, 2)$$

Р е ш е н и е .

I. Для данного примера легко определить начальную крайнюю точку $u^0 = (17, 0, 12, 0, 0, 6)$. Векторы условий, соответствующие положительным координатам крайней точки, т.е. базисные вектора – a^1, a^3, a^6. Составим симплекс таблицу

Баз	*Zб*	*0*	*6*	*-3*	5	*-2*	*-4*	2	
		A₀	*A₁*	*A₂*	*A₃*	*A₄*	*A₅*	*A₆*	θ
x_1	6	17	1	2	0	1	3	0	17/2
x_3	5	12	0	(4)	1	0	1	0	12/4
x_6	2	6	0	1	0	8	-1	1	6/1
	Δк	174	0	37	0	24	25	0	

⇧

Проверяем решение на оптимальность, для этого вычисляем Δ_k:

$$\Delta_0 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 17 \\ 12 \\ 6 \end{pmatrix} - 0 = 174, \quad \Delta_1 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - 6 = 0, \; \Delta_2 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 2 \\ 4 \\ 1 \end{pmatrix} - (-3) = 37,$$

$$\Delta_3 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - 5 = 0, \; \Delta_4 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 8 \end{pmatrix} - (-2) = 24, \; \Delta_5 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 3 \\ 1 \\ -1 \end{pmatrix} - (-4) = 25$$

$$\Delta_6 = \begin{pmatrix} 6 \\ 5 \\ 2 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - 2 = 0.$$

Решение не является оптимальным, потому что Δ2>0, Δ4>0, Δ5>0, следовательно, наибольшую Δ2>0 можно улучшить.

II. Ключевым столбцом является столбец A_2, т.к. max Δ_k =37. Ключевой строкой является строка x_3, т.к min ϴ =12/4. Ключевой элемент равен 4. Тогда x_2 входит в базис, а x_3 выходит из базиса.

Заполняем следующую симплексную таблицу.

а) переписываем ключевую строку, разделив ее на ключевой элемент;

б) заполняем базисные столбцы;

в) остальные коэффициенты таблицы находим по правилу «прямоугольника»

Правило «прямоугольника» заключается в следующем:

$нов.эл.=17-\frac{2\times 12}{4}=11$, $нов.эл.=1-\frac{0\times 0}{4}=1-0=1$, $нов.эл.=0-\frac{1\times 2}{4}=0-1/2=-1/2$

$нов.эл.=1-\frac{(-1/2)\times 0}{4}=1-0=1$, $нов.эл.=3-\frac{1\times 2}{4}=5/2$, $нов.эл.=0-\frac{0\times 2}{4}=0$,

$нов.эл.=.=6-\frac{1\times 12}{4}=6-3=3$, $нов.эл.=0-\frac{1\times 0}{4}=0-0=0$, $нов.эл.=8-\frac{1\times 0}{1}=8-0=8$,

$нов.эл.=-1-\frac{1\times 1}{4}=-1-1/4=-5/4$, $нов.эл.==1-\frac{1\times 0}{4}=1$

Считаем Δ_k:

$\Delta_0=\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}11\\3\\3\end{pmatrix}-0=63$, $\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}1\\0\\0\end{pmatrix}-6=0$, $\Delta_2=\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}0\\1\\0\end{pmatrix}-(-3)=0$,

$\Delta_3=\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}-1/2\\1/4\\-1/4\end{pmatrix}-5=-37/4$, $\Delta_4=\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}1\\0\\8\end{pmatrix}-(-2)=24$,

$\Delta_5=\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}5/2\\1/4\\-5/4\end{pmatrix}-(-4)=63/4$, $\Delta_6=\begin{pmatrix}6\\-3\\2\end{pmatrix}\times\begin{pmatrix}0\\0\\1\end{pmatrix}-2=0$

Баз	*Zб*	*0*	*6*	*-3*	*5*	*-2*	*-4*	*2*	
		A_0	*A_1*	*A_2*	*A_3*	*A_4*	*A_5*	*A_6*	*θ*
x_1	6	11	1	0	-1/2	1	5/2	0	11
x_2	-3	3	0	1	1/4	0	1/4	0	
x_6	2	3	0	0	-1/4	(8)	-5/4	1	3/8
	Δк	63	0	0	-37/4	24	63/4	0	

⇧

Решение не является оптимальным, потому что Δ4>0, Δ5>0,, следовательно Δ4>0 можно улучшить.

III. Ключевым столбцом является столбец A_4, т.к. max Δ_k =24. Ключевой строкой является строка x_6, т.к min Ө =3/8. Ключевой элемент равен 8. Заполняем следующую симплексную таблицу.

а) переписываем ключевую строку, разделив ее на ключевой элемент;

б) заполняем базисные столбцы;

в) остальные коэффициенты таблицы находим по правилу «прямоугольника»

Баз	$Z_б$	*0*	*6*	*-3*	*5*	*-2*	*-4*	*2*	
		A_0	A_1	A_2	A_3	A_4	A_5	A_6	θ
x_1	6	85/8	1	0	-15/32	0	*(85/32)*	*1/8*	4
x_2	-3	3	0	1	1/4	0	*1/4*	0	12
x_4	-2	3/8	0	0	--32/	1	-5/32	1/8	0
	Δк	54	0	0	-17/2	0	39/2	-3	

⇧

Решение не является оптимальным, потому что Δ5>0, следовательно Δ5>0 можно улучшить.

IY. Ключевым столбцом является столбец A_5, т.к. max Δ_k =39/2. Ключевой строкой является строка x_1, т.к min Ө =4. Ключевой элемент равен 8. Заполняем следующую симплексную таблицу.

а) переписываем ключевую строку, разделив ее на ключевой элемент;

б) заполняем базисные столбцы;

в) остальные коэффициенты таблицы находим по правилу «прямоугольника»

Баз	$Z_б$	*0*	*6*	*-3*	5	*-2*	*-4*	2
		A_0	A_1	A_2	A_3	A_4	A_5	A_6
x_5	-4	4	32/85	0	*-15/85*	0	1	*-4/85*
x_2	-3	2	$-\frac{8}{85}$	1	$\frac{25}{85}$	0	0	$\frac{1}{85}$
x_4	-2	1	$\frac{1}{17}$	0	$-\frac{5}{85}$	1	0	$\frac{10}{85}$
Δк		-24	$-\frac{624}{85}$	0	$-\frac{430}{85}$	0	0	$-\frac{177}{85}$

Т.к все $\Delta_k \geq 0$, то решение является оптимальным.

Ответ: min Z(X) = -24 при X = (0; 2; 0;1;4;0).

Б) В случае когда не можем обеспечить необходимым количеством базисных векторов, переходим к М задаче. Для этого добавляем искуственные переменные к целевой функции и к ограничениям. Далее применяется симплекс метод. Строка Δ_k в этом случае записивается в виде

$$\Delta k_j = \alpha_j + M\beta_j .$$

Где М – большое положительное число. Кроме того, если выводимый вектор совпадает с искусттвенным переменным, то в последующей итерации этот столбец исключается.

В) Решить задачу линейного программирования

$$\begin{aligned}
&z(x) = -2x_1 - 3x_2 + 5x_3 - 6x_4 - 4x_5 \to \min\\
&2x_1 + x_2 - x_3 + x_4 = 5\\
&x_1 + 3x_2 + x_3 - x_4 + 2x_5 = 8\\
&-x_1 + 4x_2 + x_4 = 1\\
&x_j \geq 0,\ j = \overline{1,5}
\end{aligned}$$

Соответствующая М–задача имеет вид:

$$\begin{aligned}
&z(x) = -2x_1 - 3x_2 + 5x_3 - 6x_4 - 4x_5 + Mx_6 + Mx_7 \to \min\\
&2x_1 + x_2 - x_3 + x_4 + x_6 = 5\\
&\frac{1}{2}x_1 + \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_4 + x_5 = 4\\
&-x_1 + 4x_2 + x_4 + x_7 = 1\\
&x_j \geq 0,\ j = \overline{1,7}
\end{aligned}$$

Для М–задачи матрица A, векторы b и c равны:

$$\begin{pmatrix} 2 & 1 & -1 & 1 & 0 & 1 & 0\\ 1 & 3 & 1 & -1 & 1 & 0 & 0\\ -1 & 4 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} = \left(a^1, a^2, a^3, a^4, a^5, a^6, a^7\right)$$

$$b = \begin{pmatrix} 5\\ 8\\ 1 \end{pmatrix}, \quad c' = (-2, -3, 5, -6, 4, M, M) \cdot$$

Р е ш е н и е . Начальная крайняя точка $x^0=(0,\ 0,\ 0,\ 0,\ 4,\ 5,\ 1)$, Величину $z_j - c_j$ представим в виде $z_j - c_j = \alpha_j M + \beta_j$, $j = \overline{1,n}$, т.е. вместо одной строки для $z_j - c_j$ вводятся две: в первую записываются значения β_j, во вторую – α_j. Поскольку М – достаточно большое положительное число, то

$z_j - c_j > z_k - c_k$, если $\alpha_j > \alpha_k$. Если же $\alpha_j = \alpha_k$, то $\beta_j > \beta_k$.

Базис	$\bar{c}$	B	-2	-3	5	-6	4	M	M
			a¹	a²	a³	a⁴	a⁵	a⁶	a⁷

I

A^6	M	5	2	1	-1	1	0	1	0
A^5	4	4	1/2	3/2	1/2	-1/2	1	0	0
A^7	M	1	-1	(4)	0	1	0	0	1 ⇒
$z_j - c_j$		16	4	9	-3	4	0	0	0
		6	1	5	-1	2	0	0	0

⇧

II

a^6	M	19/4	(9/4)	0	-1	3/4	0	1⇒
a^5	4	29/8	7/8	0	1/2	-7/8	1	0
a^2	M	1/4	-1/4	1	0	1/4	0	0
$z_j - c_j$		55/4	25/4	0	-3	7/4	0	
		19/4	9/4	0	-1	3/4	0	0

⇧

III

a^1	-2	19/9	1	0	-4/9	3/9	0
a^5	4	16/9	0	0	8/9	-21/18	1
a^2	-3	7/9	0	1	-1/9	3/9	0
$z_j - c_j$		5/9	0	0	-2/9	-1/3	0
		0	0	0	0	0	0

Как следует из последней симплекс–таблицы, решением М-задачи является вектор $x^* = (19/9,\ 7/9, 0, 0, 16/9, 0, 0)$. Тогда решение исходной задачи – вектор $x^* = (19/9,\ 7/9, 0, 0, 16/9, 0, 0)$, $z(x^*) = 5/9$.

1.4. Двойственность в задачах линейного программирования.

С каждой задачей линейного программирования тесно связана другая линейная задача, называемая, двойственной; первоначальная задача называется исходной, или прямой.

Задача оптимального использования сырья. Допустим, предприятие распологает сырьем вида S_1, S_2, S_3, S_4 в количестве 19, 13, 15 и 18 соответственно. Это сырье используется для производства одной единицы P_1 и P_2, продаваемой по цене 7 и 5 ден. ед. соответственно. Для производства одной единицы продукции P_1 требуется 2 единицы сырья S_1, 2 ед.$-S_2$ и 3 ед. $-S_4$. Для производства одной единицы продукции P_2

трбуется 3 единицы сырья S_1, 1 ед. $-S_2$ и 1 ед. $-S_3$. Составить такой план производства, при котором выручка от реализации пройзведенной продукции будет максимальной.

Пусть $x_1 -$ количество производимой продукции P_1 и $x_2 -$ количество производимой продукции P_2. Тогда целевая функция имеет вид:

$$f = 7x_1 + 5x_2 \to \max \tag{1.1}$$

при ограничениях:

$$\begin{cases} 2x_1 + 3x_2 \le 19, \\ 2x_1 + x_2 \le 13, \\ x_2 \le 15, \\ 3x_1 \le 18 \end{cases} \tag{1.2}$$

и условия неотрицательности:

$$x_j \ge 0; \quad j = \overline{1,n} \tag{1.3}$$

Предположим, что некоторая организация желает приобрести не продукию, а сырье, которым распологает предприятие. По какой цене эта организация стала бы покупать указанное сырье? Обазначим через y_1, y_2, y_3, y_4, цену единицы сырья вида S_1, S_2, S_3, S_4.

Выручка от продажи всего сырья, расходуемого на единицу продукции вида P_1 по ценам y_1, составит $2y_1 + 2y_2 + 0 \cdot y_3 + 3y_4$, на единицу продукции вида P_2 – $3y_1 + y_2 + y_3 + 0y_4$. Предприятие продасть сырье, если его стоимость будет превышать цену единицу продукции, получаемой с использованием этого сырья:

$$\begin{cases} 2y_1 + 2y_2 + 0 \cdot y_3 + 3y_4 \ge 7 \\ 3y_1 + 1y_2 + 3y_3 + 0 \cdot y_4 \ge 5. \end{cases} \tag{1.4}$$

С другой стороны, общая стоимость всех запасов приобретаемого сырья составить:

$$Z(y) = 19y_1 + 13y_2 + 15y_3 + 18y_4 \tag{1.5}$$

Ясно,что организация стремится приобрести сырье по возможности дешевле, то есть минимизировать функцию (1.15). Таким образом, математическая запись задачи будет иметь вид:

$$G(y) = 19y_1 + 13y_2 + 15y_3 + 18y_4 \to \min \qquad (1.6)$$

$$\begin{cases} 2y_1 + 2y_2 + 0 \cdot y_3 + 3y_4 \geq 7 \\ 3y_1 + 1y_2 + 3y_3 + 0 \cdot y_4 \geq 5 \end{cases} \qquad (1.7)$$

и при условиях отрицательности:

$$y_i \geq 0; \quad i = \overline{1,m} \qquad (1.8)$$

Таким образом, двойственная задача (1.6)-(1.8) соответствует следующей экономической проблеме: *по каким минимальным ценам следует продовать ресурсы, чтобы прибыль от их релизации была больше прибыли, полученной от реализации продукции, изготавливаемой с иcползованием этих ресурсов.*

Две задачи линейного программирования (1.11)-(1.3), (1.6)-(1.8), связанные подобного рода особенностями, называются *взаимно-двойственными* или *сопряженными.*

Общая формулировка прямой и двойственной задачи

Дадим достаточно общее определение двойственной задачи по отношению к прямой, заданной в виде:

$$f = c_1x_1 + c_2x_2 + \dots + c_nx_n \to \max \qquad (1.9)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n < R_1 > b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n < R_2 > b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n < R_n > b_n. \end{cases} \qquad (1.10)$$

$$x_1, x_2, \dots, x_n \geq 0. \qquad (1.11)$$

Здесь: $f = c_1x_1 + c_2x_2 + \dots + c_nx_n$ – целевая функция; $R_i, i = \overline{1,m}$ – операции отношения =, ≥, ≤; b_i– заданные константы. Очевидно, что любую задачу линейного программирования можно представить в одной из двух указанных форм.

Тогда двойственной по отношению к задаче (1.9)−(1.11) будем называть задачу:

$$g = b_1y_1 + b_2y_2 + \dots + b_my_m \to \min \qquad (1.12)$$

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + ... + a_{m1}y_m \ge c_1 \\ a_{12}y_1 + a_{22}y_2 + ... + a_{m2}y_m \ge c_2 \\ .... \\ a_{1n}y_1 + a_{2n}y_2 + ... + a_{mn}y_m \ge c_n. \end{cases} \tag{1.13}$$

и, если в прямой задаче $R_i = (\le), i = \overline{1,m}$, то

$$y_1, y_2, ..., y_m \ge 0. \tag{1.14}$$

Правила построения двойственной задачи:

а) целевая функция исходной задачи (1.9)−(1.11) задается на максимум, а целевая функция двойственной задачи (1.12)−(1.14) – на минимум;

б) матрица системы сильных ограничений двойственной задачи получится из матрицы системы ограничений прямой задачи транспонированием;

в) число переменных двойственной задачи равно числу сильных ограничений прямой, а число ограничений в двойственной задаче равно числу переменных прямой;

г) коэффициентами при неизвестных в целевой функции двойственной задачи являются свободные члены системы сильных ограничений прямой задачи, а правыми частями системы сильных ограничений двойственной задачи – коэффициенты при неизвестных в целевой функции исходной задачи.

Отметим, что если все $R_i = (\le), i = \overline{1,m}$ и, следовательно, в двойственной задаче выполнены условия (1.14) – переменные неотрицательны, то пара задач (1.9)−(1.11) и (1.12)−(1.14) называется симметричной.

Свойства двойственной задачи. Приведем теоремы, устанавливающие зависимости между решениями прямой и двойственной задач.

Теорема 1.1. *Значение целевой функции задачи (1.9)–(1.11) при любом допустимом плане **x** не превышает значения целевой функции задачи (1.12)–(1.14) при любом ее допустимом плане y: $f(x) \le g(y)$. Если для каких-нибудь **x** и **y** значения целевых функций совпадают, т.е. $f(x) = g(y)$, то **x** оптимальное решение (1.9)–(1.11), **y** – оптимальное решение (1.12)–(1.14).*

◄ Пусть $x = (x_1, x_2, ..., x_n)$ – допустимое решение задачи (1.9)–(1.11), $y = (y_1, y_2, ..., y_m)$ – допустимое решение задачи (1.12)–(1.14). Подставим решение ***x*** в ограничения (1.10), умножим первое равенство на y_1, второе на y_2,…, m–е на y_m и сложим полученные равенства:

$$(a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n)y_1 + (a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n)y_2 +$$
$$+...(a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n)y_m = b_1y_1 + b_2y_2 + ... + b_my_m.$$

Перегруппируем слагаемы в левой части:

$$(a_{11}y_1 + a_{21}y_2 + ... + a_{m1}y_m)x_1 + (a_{12}y_1 + a_{22}y_2 + ... + a_{m2}y_m)x_2 +$$
$$+...(a_{1n}y_1 + a_{2n}y_2 + ... + a_{mn}y_n)y_m = b_1y_1 + b_2y_2 + ... + b_my_m. \quad (1.15)$$

Так как y – допустимое решение задачи (1.12)–(1.14), то для любого $j = \overline{1,n}$ имеем: $a_{1j}y_1 + a_{2j}y_2 + ... + a_{mj}y_n \ge c_j$.Заменяя в (1.15) суммы, заключенные в скобки, соответствую-щими c_j и учитывая (1.10), получаем:

$c_1x_1 + c_2x_2 + ... + c_nx_n \le b_1y_1 + b_2y_2 + ... + b_my_m$

или, что тоже самое $f(\boldsymbol{x}) \le g(\mathbf{y})$. Первая часть теоремы доказана.

Пусть для каких-нибудь $\boldsymbol{x}$,$\mathbf{y}$ имеем: $(c, x') > (b^T, y)$. Допустим от противного, что $\boldsymbol{x}$ не является оптимальным решением задачи (1.9)-(1.11). Это значит, что есть такое другое допустимое решение $\boldsymbol{x}'$задачи (1.9)–(1.11), что $(c, x') > (c, x') = (b^T, y)$а это противоречит первой части данной теоремы. Следовательно, при нашем условии $\boldsymbol{x}$ – оптимальное решение задачи (1.9)–(1.11). Аналогично показывается, что $\mathbf{y}$ – оптимальное решение задачи (1.12)–(1.14).►

Теорема 1.2. *Если одна из пар двойственных задач (1.9)–(1.11) или (1.12)– (1.14) имеет оптимальное решение, то и другая также имеет оптимальное решение, причем их оптимумы совпадают (max f = min g). Если целевая функция одной из пары двойственных задач не ограничена (для прямой – сверху, а для двойственной − снизу), то допустимая область другой задачи пуста.*

◄Ввиду взаимной двойственности задач (1.9)-(1.11) и (1.12)– (1.14) достаточно провести доказательство, принимая за исходную одну из задач. Пусть, например, задача (1.9)–(1.11) имеет оптимальные решения, и $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$– одно из них, найденное симплекс-методом, и $(A_{i_1}, A_{i_2}, ..., A_{i_m})$– его базис. Тогда на последней итерации имеем (j=1,2,…,n) такчто zj ≥ cj, j=1,2,…,n, или, в векторной форме

$(z_1, z_2, ..., z_n) \ge (c_1, c_2, ..., c_n).$(1.16)

При рассмотрении симплекс-метода мы выяснили, что вектор$(z_1, z_2, ..., z_n)$ может быть найден по формуле$(z_1, z_2, ..., z_n) = (c_{i_1}, c_{i_2}, ..., c_{i_m})B^{-1}A = c_B B^{-1}A,$ где B^{-1} – обратная матрица к базисной матице$B = (A_{i_1}, A_{i_2}, ..., A_{i_m})$, A – матрица системы ограничений –уравнений задачи: $A = (A_1, A_2, ..., A_n)$. Следовательно , из (1.16) имеем

$$(c_{i_1}, c_{i_2}, ..., c_{i_m})B^{-1}A \ge (c_1, c_2, ..., c_n) = c \quad (1.17)$$

Введем обозначение $(c_{i_1}, c_{i_2}, ..., c_{i_m})B^{-1} = y = (y_1, y_2, ..., y_m)$. Тогда неравенство (1.17) принимает вид $YA \geq c$. Докажем,что y – оптимальное решение задачи (1.12)– (1.14). Имеем:

$$(y,b) = (c_{i_1}, c_{i_2}, ..., c_{i_m})B^{-1}b = (c_{i_1}, c_{i_2}, ..., c_{i_m})(x_1^{(0)}, x_2^{(0)}, ..., x_m^{(0)}) =$$
$$= c_{i_1}x_1^{(0)} + c_{i_2}x_2^{(0)} + ... + c_{i_m}x_m^{(0)} = (c, x^{(0)})$$

Таким образом, значение целевой функции задачи (1.12)– (1.14) при её допустимом решении ***y***совпадает с значением целевой функции задачи (1.9)–(1.11) при её допустимом решении . Согласно теореме 1 это означает, что **y** оптимальное решение задачи (1.12)– (1.14) Первая часть теоремы доказана. Докажем вторую её часть. Пусть известно, что целевая функция, например, задачи (1.9)– (1.11) не ограничена сверху на допустимом множестве. Допустим, от противного, что задача (1.12)– (1.14)) имеет при этом допустимое решение, и пусть **y** – одно из них. Ввиду неограниченности целевой функции задачи (1.9)–(1.11) найдётся такое допустимое её решение $(c,x) > (b,y)$. Что противоречить теореме1.▶

1.4.1. Решения двойственных задач.

А) Решение несимметричных двойственных задач.

Исходная задача. Найтиминимальное значение линейной функции

$$f = x_2 - x_4 - 3x_5 \to \min$$

при ограничениях

$$\begin{cases} x_1 + 3x_2 - x_4 + x_5 = 1, \\ -x_2 + x_3 + 2x_4 - x_5 = 2, \\ 3x_2 + x_5 + x_6 = 5 \end{cases} x_j \geq 0; \quad j = \overline{1,6}.$$

Здесь

$$c = (0;1;0;-1;-3;0),$$

$$A = \begin{pmatrix} 1 & 2 & 0 & -1 & 1 & 0 \\ 0 & -4 & 1 & 2 & -1 & 0 \\ 0 & 3 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad A'' = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -4 & 3 \\ 0 & 1 & 0 \\ -1 & 2 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Двойственная задача. Найти максимальное значение линейной функции

$$g = y_1 + 2y_2 + 5y_3 \to \max$$

при ограничениях

$$\begin{cases} y_1 \le 0, \\ 2y_1 - 4y_2 + 3y_3 \le 0, \\ y_2 \le 0 \\ -y_1 + 2y_2 \le -1, \\ y_1 - y_2 + y_3 \le -3, \\ y_3 \le 0 \end{cases}$$

Решение исходной задачи находим симплекс методом (табл. 1.1).

Оптимальный план исходной задачи $X^* = (0;\frac{1}{3};0;\frac{11}{3};4;0)$ при котором $f_{\min} = -\frac{46}{3}$, получен в четвертой итерации табл. 1.1.

Таблица 1.1

№№	Базис	C бас-зиса		0	1	0	-1	-3	0
			A₀	**A₁**	**A₂**	**A₃**	**A₄**	**A₅**	**A₆**
1	A_1	0	1	1	2	0	-1	[1]	0
2	A_3	0	2	0	-4	1	2	-1	0
3	A_6	0	5	0	3	0	0	1	0
m+1	$z_j - c_j$		0	0	-1	0	1	**3**	0
1	A_5	-3	1	1	2	0	-1	1	0
2	A_3	0	3	1	-2	1	[1]	0	0
3	A_6	0	4	-1	1	0	1	0	0
m+1	$z_j - c_j$		-3	-3	-7	0	4	**0**	0
1	A_5	-3	4	2	0	1	0	1	0
2	A_4	-1	3	1	-2	1	1	0	0
3	A_6	0	1	-2	[3]	-1	0	0	1
m+1	$z_j - c_j$		-15	-7	1	-4	0	**0**	0
1	A_5	-3	4	2	0	1	0	1	0
2	A_4	-1	11/3	-1/3	0	1/3	1	0	2/3
2/33	A_2	0	1/3	-2/3	1	-1/3	0	0	1/3
m+1	$z_j - c_j$		-46/3	-19/3	0	-11/3	0	0	-1/3

Используя эту итерацию, найдем оптимальный план двойственной задачи . Согласно теореме двойствености оптимальный план двойственной задачи находится из соотношения $Y^* = C^* D^{-1}$, где матрица D^{-1} – матрица, обратная матрице, составленой из компонент векторов, входящих в последний базис, при котором получен оптимальный план исходной задачи. В последний базис входят векторы A_5, A_4, A_2; значит,

$$D = \left(A_5, A_4, A_2\right) = \begin{pmatrix} 1 & -1 & 2 \\ -1 & 2 & -4 \\ 1 & 0 & 3 \end{pmatrix}$$

Обратная матрица D^{-1} образованная из коэффициентов, стоящих в столбцах A_1, A_3, A_6 четвертой итерации:

$$D^{-1} = \begin{pmatrix} 2 & 1 & 0 \\ -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{-2}{3} & -\frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

Из этой же итерации следует, что $C^* = \left(-3; -1; 1\right)$. Таким образом

$$Y^* = C^* D^{-1} = \left(-3; -1; 1\right) \cdot \begin{pmatrix} 2 & 1 & 0 \\ -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{-2}{3} & -\frac{1}{3} & \frac{1}{3} \end{pmatrix},$$

$$Y^* = \left(-\frac{19}{3}; \frac{-11}{3}; \frac{-1}{3}\right), \text{ т .е. } Y^* = C^* X_i,$$

где X_i – коэффициенты разложения последней итерации, стоящие в столбцах веторов пеервоначального единичного базиса. Итак, двойственную переменную можно получить из значения оценки (m+1)-й строки, стоящей против соответствующего вектора, входившего в первоначальный базис, если к ней прибавить соответствующее значение коэффициента линейной функции:

$$y_1 = -\frac{19}{3} + 0 = -\frac{19}{3}; \;\; y_2 = -\frac{11}{3} + 0 = -\frac{11}{3}; \; y_3 = -\frac{1}{3} + 0 = -\frac{1}{3}.$$

При этом плане $g_{\max} = -\dfrac{46}{3}$.

Б) Решение симметричных двойственных задач.

Исходная задача. Найти минимальное значение линейной функции

$$f = x_1 + 2x_2 + 3x_3 \to \min$$

при ограничениях

$$\begin{cases} 2x_1 + 2x_2 - x_3 \ge 2, \\ x_1 - x_2 - 4x_3 \le -3, \\ 2x_1 + x_2 - 2x_3 \ge 6, \\ 2x_1 + x_2 - 2x_3 \ge 3, \\ x_j \ge 0 \ \ (j = 1,2,3) \end{cases}$$

Очевидно, для того чтобы записать двойственную задачу второе неравенство умножим на -1.

Двойственная задача. Найти максимум линейной функции

$$g = 2y_1 + 3y_2 + 6y_3 + 3y_4 \to \max$$

при ограничениях

$$\begin{cases} 2y_1 - y_2 + y_3 + 2y_4 \le 1, \\ 2y_1 + y_2 + y_3 + y_4 \le 2, \\ -y_1 + 4y_2 - 2y_3 - 2y_4 \le 3, \\ y_3 \le 0, \ (i = 1,2,3,4). \end{cases}$$

Для решения исходной задачи необходимо ввести четыре дополнительные переменные и после преобразования системы – одну искусственную. Таким образом, исходная симплекс-таблица будет состоять из шести строк и девяти столбцов, элементы которых подлежать преобразованию. Для решения двойственной задачи необходимо ввести три дополнительные переменные. Система ограничений не требует предварительных преобразований, ее первая симплекс-таблица четыре строки и восемь столбов.
Двойственную задачу решаем симплекс методом (табл. 1.2)

Таблица 1.2

i	Базис	С базиса	A_0	2	3	6	3	-3	0	0
				A_1	A_2	A_3	A_4	A_5	A_6	A_7
1	A_5	0	1	2	-1	[1]	2	1	0	0
2	A_6	0	2	2	1	1	-1	0	1	0
3	A_7	0	3	-1	4	-2	-2	0	0	1
$m+1$	$Z_j - C_j$		0	-2	-3	- 6	-3	0	0	0
1	A_3	6	1	2	-1	1	2	1	0	0
2	A_6	0	1	0	[2]	0	-1	-1	1	0

3	A_7	0	5	3	6	0	2	2	0	1
$m+1$	$Z_j - C_j$		6	10	-9	0	9	6	0	0
1	A_3	6	3/2	2	0	1	3/2	1/2	1/2	0
2	A_2	3	1/2	0	1	0	-1/2	-1/2	1/2	0
3	A_7	1	2	3	0	0	4	5	3	1
$m+1$	$Z_j - C_j$		21/2	-19/3	0	0	9/2	3/2	9/2	0

Оптимальный план двойственной задачи $Y^* = (0;1/2;\ 3/2;0)$, $g_{\max} = 21/2$.

Оптимальный план исходной задачи находим, используя оценки $m+1$-й строки последней итерации, стоящие в столбцах A_5, A_6, A_7:
$x_1 = \dfrac{3}{2} + 0 = \dfrac{3}{2};\ x_2 = \dfrac{9}{2} + 0 = \dfrac{9}{2};\ x_3 = 0 + 0 = 0.$ При оптимальном плане исходной задачи $X^* = (3/2;\ 9/2;0)$ линейная функция достигает наименьшего значения $f_{\min} = 21/2$

1.5. Двойственный симплекс метод.

Ранее мы показали, что для получения решения исходной задачи можно перейти к двойственной и, используя оценки ее оптимального плана , определить оптимальное решение исходной задачи.

Переход к двойственной задаче не объзателен, так как если рассмотреть первую симплексную таблицу с единичным дополнительным базисом, то легко заметить, что в столбцах записана исходная задача, в стороках – двойственная. Причем оценками плана исходной задачи являются C_j, а оценками плана двойственной задачи – b_i. Решим двойственную задачу со симплексной таблице, в которой записана исходная задача; найдем оптимальный план двойственной задачи, а вместе с ним и оптимальный план исходной задачи. Этот метод носит название ***двойственного симплекс метода.***

Пусть необходимо решить исходную задачу линейного программирования, поставленную в общем виде: минимизировать функцию $f(x) = C \cdot X$ при $A \cdot X = A_0,\ X \geq 0$. Тогда в двойственной задаче необходимо максимизировать функцию $g(y) = Y \cdot A_0$ при $Y \cdot A \leq C$. Допустим, что выбран такой базис $D = (A_1, A_2, ..., A_l, ..., A_m)$, при котором хотя бы из компонент $X = D^{-1} \cdot A_0 = (x_1, x_2, ..., x_l, ..., x_m)$ отрицательная (например $x_l < 0$), но для всех векторов A_j выполняется соотношение $Z_j - C_j \leq 0$ ($j = \overline{1,n}$). Тогда на основании теоремы двойственности $E = C_{баз} D^{-1}$ – план двойственной задачи. Этот план не оптимальный, так как , с одной стороны, при выбранном базисе X содержит отрицательную компоненту и не является планом исходной задачи, с другой стороны, оценки оптимального плана двойственной задачи должны быть не отрицательными.

Таким образом, вектор A_l, соответствующий компоненте $x_l < 0$, следует исключить из базиса исходной задачи, а вектор, соответствующий отрицательной оценке,– включить в базис двойственной задачи.

Для выбора вектора, включаемого в базис исходной задачи, просматриваем l-ю сторку: если в ней не содержаться $x_{lj} < 0$, то линейная функция двойственной задачи не ограничена на многограннике решений, а исходная задача не имеет решений. Если же некоторые $x_{lj} < 0$, то для столбцов, содержащих эти отрицательные значения, вычисляем $\theta_{0j} = \min\left(x_i / x_{lj}\right) \geq 0$ и определяем вектор, соответствующий $\max \theta_{0j}\left(Z_j - C_j\right)$ при решении исходной задачи на минимум и $\min \theta_{0j}\left(Z_j - C_j\right)$ при решении исходной задачи на максимум. Этот вектор и включаем в базис исходной задачи. Вектор, который необходимо исключить из базиса исходной задачи, определяется направляющей строкой.

Если $\theta_{0j} = \min\left(x_i / x_{lj}\right) = 0$, т.е. $x_i = 0$, то x_{lj} берется за разрешающий элемент только в том случае, если $x_{lj} > 0$. Такой выбор разрешающего элемента на данном этапе не приводит к увеличению количества отрицательных компонент вектора X. Процесс продолжаем до получения $X \geq 0$; при этом находим оптимальный план двойственной задачи, следовательно, и оптимальный план исходной задачи.

В процессе вычислений по алгоритму двойственного симплексного метода условие $Z_j - C_j \leq 0$ можно не учытывать до исключение всех $x_i < 0$, затем оптимальный план находится обычным симплекс методом. Это удобно использовать, если все $x_l < 0$; тогда для перехода к плану исходной задачи за одну итерацию необходимо θ_{0j} определить не по минимуму, а по максимуму отношений, т.е. $\theta_{0j} = \max\left(x_i / x_{lj}\right) > 0$.

Двойственным симплексным методом можно решать задачи линейного программирования, системы ограничений которых при положительном базисе содержат свободные члены любогог знака. Этот метод позволяет уменьшить клоичество преобразований системы ограничений, а также размеры симплексной таблицы.

Пример 1.1. Найти минимальное значение функции

$$f(x) = -2x_1 + x_2 + 5x_3 \rightarrow \min \ ,$$

при ограничениях

$$\begin{cases} x_1 + x_2 - x_3 \le 4, \\ x_1 - 5x_2 + x_3 \ge 5, \\ x_j \ge 0 \ (j = \overline{1,3}). \end{cases}$$

Приведем двойственную задачу к каноническому виду:

$$f(x) = -2x_1 + x_2 + 5x_3 \to \min ,$$

$$\begin{cases} x_1 + x_2 - x_3 + x_4 = 4, \\ -x_1 + 5x_2 - x_3 + x_5 = -5, \\ x_j \ge 0 \ (j = \overline{1,5}). \end{cases}$$

Составим первую симплексную таблицу, выразив за базис веторов A_4 и A_6 .

Базисные переменные	$C_Б$	A_0	7	20	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_4	0	4	1	1	-1	1	0
A_5	0	-5	-1	5	-1	0	1
$z_j - c_j$		0	2	-1	-5	0	0

Так как $x_2 = -5 < 0$, то просматриваем клэффициенты второй строки. Среди них два отрицательных коэффициента, стоящие в столбцах соответствующих векторам A_1 и A_3 . Имеем

$$\theta_{01} = \min\left(\frac{4}{1}; \frac{-5}{-1}\right) = \frac{4}{1}, \ \theta_{01}(z_1 - c_1) = \frac{4}{1} \cdot 2 = 8 , \ \theta_{03} = \frac{-5}{-1} = 5, \ \theta_0(z_3 - c_3) = 5(-5) = -25 .$$

Исходная задача решается на отыскание минимального значения линейной функции, поэтому в базис исходной задачи надо включать вектор, которому $\max \theta_{0j}(z_j - c_j) = \max(-25; 8) = 8$, т.е. вектор A_1 с разрешающим элементом 1, вектор A_4 исключим из базиса.В результате одного полного исключения получаем таблицу,

Базисные переменные	$C_Б$	A_0	7	20	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_1	-2	4	1	1	-1	1	0
A_5	0	-1	0	6	-3	1	1
$z_j - c_j$		-8	0	-3	-3	-2	0

A_1	-2	9/2	1	-2	0	1/2	-1/2
A_3	5	1/2	0	-3	1	-1/2	-1/2
$z_j - c_j$		-13/2	0	-12	0	-7/2	-3/2

где в последней итерации найден оптимальный план исходной задачи:

$$X^* = (9/2; 0; 1/2),\ Z_{\min} = -13/2$$

и двойственной задачи:

$$Y^* = (7/2; 3/2),\ g_{\max} = -13/2\ .$$

Для получения плана двойственной задачи было необходимо $Z_j - C_j$, соответствующие векторам перваначального базиса, умножить на -1, так как исходная и двойственная задачи симметричны.

Пример 1.2. Найти минимальное значение функции

$$f(x) = 3x_1 + 2x_2 - 4x_3 \to \min\ ,$$

при ограничениях

$$\begin{cases} x_1 + x_2 - 2x_3 \geq 4, \\ 3x_1 + x_2 - 4x_3 \geq 7, \\ x_j \geq 0\ (j = \overline{1,3}). \end{cases}$$

Приведем двойственную задачу к каноническому виду:

$$f(x) = -2x_1 + x_2 + 5x_3 \to \min\ ,$$

$$\begin{cases} -x_1 - x_2 + 2x_3 + x_4 = -4, \\ -3x_1 - x_2 + 4x_3 + x_5 = -7, \\ x_j \geq 0\ (j = \overline{1,5}). \end{cases}$$

Решение приведено в таблице

Базисные переменные	$C_Б$	A_0	7	20	0	0	0
			A_1	A_2	A_3	A_4	A_5
A_4	0	-4	-1	-1	2	1	0
A_5	0	-7	-3	-1	4	0	1
$z_j - c_j$		0	-3	-3	4	0	0

A_1	3	4	1	1	-2	-1	0
A_5	0	6	0	2	-2	-3	1
$z_j - c_j$		12	0	1	-2	-3	0
A_1	3	3/2	1	0	-1	1/2	-1/2
A_2	2	5/2	0	1	-1	-3/2	1/2
$z_j - c_j$		19/2	0	0	-1	-3/2	-1/2

Причем разрешающий элемент в первой итерации выбран по $\theta_{0j} = \max\left(x_i / x_{lj}\right) > 0$, что позволило во второй итерации перейти к опорному плану исходной задачи и в дальнейшем решать задачу обычным симплекс методом.

Оптимальный план исходной задачи $X^* = (3/2; 5/2; 0)$, $Z_{\min} = 19/2$.

Задачи симметричные, поэтому оптимальный план двойственной задачи $Y^* = (3/2; 1/2)$, $g_{\max} = 19/2$.

Контрольные вопросы по главе 1

1. *Математическая модель линейного программирования.*
2. *Стандартная и каноническая форма записи.*
3. *Симплекс метод:*
   а) ключевой столбец
   б) ключевая строка
   в) правила «прямоугольника»
4. *Задача оптимального распределения сырья*
5. *Общая формулировка прямой и двойственной задачи.*
6. *Свойства двойственной задачи.*
7. *Правила построения двойственной задачи.*
8. *Двойственный симплекс метод.*
9. *Свойства двойственной задачи.*
10. *Стандартная и каноническая форма задач линейного программирования.*
11. *Общая формулировка прямой и двойственной задачи.*
12. *Правила построения двойственной задачи.*

2. ЭЛЕМЕНТЫ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

2.1. Целочисленное линейное программирование.

Математическая модель задачи целочисленного линейного программирования имеет вид:

$$f = \sum_{j=1}^{n} c_j x_j \to \max(\min) \qquad (2.1)$$

$$\left\{ \sum_{j=1}^{n} a_{ij} x_j = b_i, \; i = \overline{1,m} \right. \qquad (2.2)$$

$$x_j, \quad j = \overline{1,n} \qquad (2.3)$$

$$x_j \in Z, \quad j = \overline{1,n} \qquad (2.4)$$

Здесь Z – множество целых чисел. В отличие от общей задачи линейного программирования оптимальный план не обязательно будет в вершине многогранника (многоугольной области) планов.

Задачи целочисленного линейного программирования при помощи различных методов.

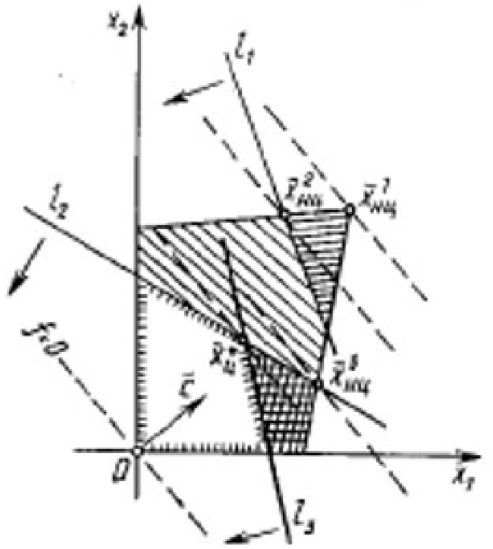
2.1.1. Метод Гомори

Общая идея этого заключется в следующем. В начале симпелкс-методом решается задача (2.1) –(2.4), т.е. без условия целочисленности переменных. Если полученный оптимальный план выражается целыми числами, то он будет решением всей задачи (2.1) –(2.4). Если же среди координат найденного оптимального плана имеется нецелые числа, то к ограничениям задачи присоединяется дополнительное линейное органичение, формируемое по специальному правилу, которому не удовлетворяет найденный нецелочисленный оптимальный план, но удовлетворяет любой целочисленный план. Говорят, что такое дополнительное органичение отсекает найденный нецелочисленный оптимальный план. Это дополнительное ограничение называют ***правильным отсечением.*** Полученную расширенную задачу вновь решают симплекс-методом. Если новый оптимальный план выражается целыми числами, то задача (2.1) –(2.4) решена. В противном случае строится следующая расширенная задача и т.д.

Геометрически каждому дополнительному ограничению соответствует гиперплоскость, отсекающая от многогранника планов некоторую его часть вместе с нецелочисленной оптимальной вершиной. Через конечное число отсечений искомая целочисленная точка сначала окажется на грани, а затем станет вершиной неоднократно усеченного многогранника планов. На рис.1.1 производится геометрическая иллюстрация для случая $n=2$. На нем изображена область планов исходной задачи и отмечена оптимальная нецелочисленная вершина $\bar{x}_{нц}^{1}$. Ближайщея к ней точка области планов с целыми координатмаи обозначена $\bar{x}_{ц}^{*}$. На рисунке показаны три прямые l_1, l_2 и l_3 соответствующие трем дополнительным ограничениям. Каждая из этих прямых отсекает часть многоугольника вместе с нецелочисленной вершиной. Так, после отсечение прямой l_1 части многоугольника планов в усеченной области оптимальной становится вершина $\bar{x}_{нц}^{2}$, после второго отсечения прямой l_2 оптимальной будет вершина $\bar{x}_{нц}^{3}$ и, наконец после третьего отсечсения прямой l_3 находится оптимальная целочисленная точка $\bar{x}_{ц}$.

1. Симплекс-методом находят оптимальный план задачи (2.1)-(2.4).

$X=\{x_1=b_1, x_2=b_2, ..., x_i=b_i, ..., x_m=b_m, ..., x_n=0\}$. Если все b_i целые, то план является оптимальным и для исходной задачи (2.1)-(2.4). Если среди b_i есть не целые, то переходят к п.2 алгоритма.



2. Среди не целых b_i выбирают, например, тот, который имеет наибольшую дробную часть. Дробной частью $\{a\}$ числа a называют разность между этим числом и его целой частью $[a]$, т.е. наиболшим целым, не превосходящим a. Например, если $a=3\frac{2}{5}$, то $\left\{3\frac{2}{5}\right\}=3\frac{2}{5}-\left[3\frac{2}{5}\right]=3\frac{2}{5}-3=\frac{2}{5}$; если $a=-5{,}7$ то $\{-5{,}7\}=-5{,}7-[-5{,}7]=-5{,}7-(-6)=0{,}3$.

В примере с отрицательными числами целая часть равна -6, так как целая часть – это ближайшее меньшее число , т.е. происходит округление до целого в меньшую сторону.

Новое ограничение (сечение) составляется на основе строки, которой соответствует дробное число b_i, и будет иметь вид:

$$\sum_{t=m+1}^{n} \alpha_{tj} x_j \geq \beta_t . \tag{2.5}$$

Здесь α_{tj} – это дробная часть коэффициентов $a_{tj} : \alpha_{tj} = \{a_{tj}\},\ j = \overline{m+1, n};$

β_t – это дробная часть числа $b_t : \beta_t = \{b_t\}.$

Для того, чтобы привести вновь составленную систему ограничений к каноническому виду, требуется преоброзовать неравенство в равенство. Введем фиктивную переменную x_{n+1}:

$$a_{tm+1}x_{m+1} + a_{tm+2}x_{m+2} + ... + a_{tn}x_n - x_{n+1} = \beta_t,\ x_{n+1} \geq 0.$$

3. Составленную расширенную задачу вновь решают симплекс-методом. Если оптимальный план будет целочисленным, то и он станет решением исходной задачи (4.1)-(4.4). Противном случае возвращается к п.2 алгоритма.

Если задача разрешима в целых числах, то через конечное число итераций оптимальный целочисленный план план будет найден.

Пример 2.1. Решить задачу: $f(x) = 3x_1 + 2x_2 \to \max$

$$\begin{cases} 5x_1 + 2x_2 \leq 20, \\ x_1 + x_2 \leq 5, \\ x_1, x_2 \geq 0, x_1, x_2 \in Z. \end{cases}$$

После выполнения этапа 1 при решении задачи симплекс-методом получилась таблица

Базис	СБ	A0	-3	-2	0	0
			A_1	A_2	A_3	A_4
x_3	0	20	5	2	1	0
x_4	0	5	1	1	0	1
$z_j - c_j$		0	3	2	0	0
x_1	3	4	1	0,4	0,2	0
x_4	0	1	0	0,6	-0,2	1
$z_j - c_j$		-12	0	0,8	-0,6	0
x_1	3	10/3	1	0	1/3	-2/3
x_2	2	5/3	0	1	-1/3	5/3
$z_j - c_j$		-40/3	0	0	-1/3	-4/3

В результате решения получился следующи ответ: $x_1 = \frac{10}{3} = 3\frac{1}{3}$, $x_2 = \frac{5}{3} = 1\frac{2}{3}$, $f = \frac{40}{3} = 13\frac{1}{3}$.

Как видно, в ответе присутствуют два дробных числа: $3\frac{1}{3}$ и $1\frac{2}{3}$. Применим метод Гомори для получения целочисленного решения.

Во первых, выберем число с максимальной дробной частью: $\left\{3\frac{1}{3}\right\}=\frac{1}{3}$ и $\left\{1\frac{2}{3}\right\}=\frac{2}{3}$, $\max\left\{\frac{1}{3};\frac{2}{3}\right\}=\frac{2}{3}.$

Это означает, что сечение будет составлено на основе строки, соответствующей числу $1\frac{2}{3}$.

Во вторых составим ограничение. Для этого выпишем линейную комбинацию чисел из строки, соответствующей числу $1\frac{2}{3}$, и переменных x_1, x_2, x_3 и x_4: $0\cdot x_1+1x_2-\frac{1}{3}x_3+\frac{5}{3}x_4\ge\frac{5}{3}.$

Тогда из формулы (15.5) получим $\frac{2}{3}x_3+\frac{2}{3}x_4\ge\frac{2}{3}$, упрощая имеем $x_3+x_4\ge 1.$ Приведем новое ограничение к каноническому виду:

$$-x_3-x_4\le -1,$$
$$-x_3-x_4+x_5=-1,\ x_5\ge 0.$$

Добавим новое сечение к системе ограничении задачи:

$$f(x)=3x_1+2x_2\to\max$$
$$\begin{cases}5x_1+2x_2+x_3=20,\\ x_1+x_2+x_4=5,\\ -x_3-x_4+x_5=-1,\\ x_1,x_2,x_3,x_4,x_5\ge 0, x_1,x_2\in Z.\end{cases}$$

Решив новую задачу симплекс методом, целочисленное решение: $x_1=3$, $x_2=2,\ f=13.$

2.1.2. Метод «Ветвей и границ»

Метод «ветвей и границ» – это метод направленного перебора множества вариантов решений задачи.

Алгоритм метода «ветвей и границ»

1. Строятся вершины первого уровня. Для каждой вершины Подсчитывается оценка нижней (верхней) границы. Вервиться вершина, которая соответствует лучшая (максимальная или минимальная) оценка.

2. Для всех вершин i –го уровня $(i\ge 2)$ подсчитывается оценка. Ветвиться та из висячих вершин уровня $i, i-1, i-2,...,1$, который соответствует лучшая (максимальная или минимальная)оценка.

3. Действие п.2 повторяется до тех пор, пока не будет получено точное решение на последнем уровне, дающее значение целевой функции f^*. Если это значение f^* не хуже оценок оставшихся висячих вершин, то найдено оптимальное решение. Если это значение f^* строго лучше, то оптимальное решение единственно.Если значенеие целевой функции f^* для последногоуровня не лучше значения оценок оставшихся висячих вершин, то переходят к п.2.

Рассмотрим задачу целочисленного линейного программирования (4.1) –(4.4). Пусть в результате решения задачи получили ответ

$$X = \{x_1 = b_1, x_2 = b_2, ..., x_m = b_m, x_{m+1} = 0, ..., x_n = 0\}$$

где b_i – дробное число. Введем два дополнительных условия:

$x_t \le z_t$ и $x_t \ge z_t + 1$, где $[x_t] \ge z_t$. Два последних неравенства разбивают область допустимых решений для переменной x_t на две подобласти. При этом из рассмотрения исключается та подобласть, в которой не содержатся целочисленные значения x_t .

Рис.15.2 Схематическое изображение метода «ветвей и границ»

Пример 2.2. Рассмотрим задачу из предыдушего примера, решенную методом Гомори:

$$f(x) = 3x_1 + 2x_2 \to \max$$

$$\begin{cases} 5x_1 + 2x_2 \le 20, \\ x_1 + x_2 \le 5, \\ x_1, x_2 \ge 0, x_1, x_2 \in Z. \end{cases}$$

В результате решения получился следующий ответ:

$x_1 = \frac{10}{3} = 3\frac{1}{3}$, $x_2 = \frac{5}{3} = 1\frac{2}{3}$, $f = \frac{40}{3} = 13\frac{1}{3}$. Как в методе Гомори, требуется выбрать число с максимальной дробной частью. Ранее было подсказано, что это x_2 этого определяем целую часть числа $\frac{5}{3}$: $\left[\frac{5}{3}\right] = \left[1\frac{2}{3}\right] = 1$. Тогда $x_2 \le 1$ и $x_2 \ge 2$.

На основе ветвления получим две новые задачи :

Задача 1

$$f(x) = 3x_1 + 2x_2 \to \max$$
$$\begin{cases} 5x_1 + 2x_2 \le 20, \\ x_1 + x_2 \le 5, \\ x_1 \ge 0, \\ 0 \le x_2 \le 1 \\ x_1, x_2 \in Z. \end{cases}$$

Задача 2

$$f(x) = 3x_1 + 2x_2 \to \max$$
$$\begin{cases} 5x_1 + 2x_2 \le 20, \\ x_1 + x_2 \le 5, \\ x_1 \ge 0, \\ x_2 \ge 2 \\ x_1, x_2 \in Z. \end{cases}$$

В результате решения задач 1 и 2 получим следующие ответы:

Задача 1

$x_1 = 3{,}6; x_2 = 1;\ f = 12{,}8$

задача 2

$x_1 = 3; x_2 = 2;\ f = 13.$

Изобразим схематически ветвление в решенной задаче

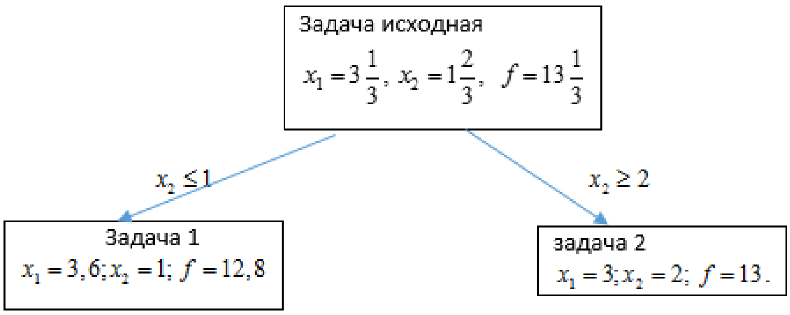


Рис. 15.3. Ветвление задачи

Таким образом, с помощью двух различных методов получили одно решение задачи целочисленного программирования.

Контрольные вопросы глава 2

1. *Дополнительное линейное ограничение.*
2. *Правильное отсечение*
3. *Дробная часть числа.*
4. *Фиктивная переменная.*
5. *Алгоритм метода «ветвей и границ».*

3. ВВЕДЕНИЕ В ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

3.1. Краткая история дисциплины «Исследование операций»

Исследование операций (ИСО) – это наука, занимающаяся разработкой и практическим применением методов оптимального управления организационными системами.

Предмет ИСО – это системы организационного управления, которые характеризуются своим составом, структурой, связями и органами управления. Организация состоит из множества взаимодействующих элементов: люди, машины (оборудование, компьютеры и др.), материальные и финансовые средства. Организация имеет свою структуру, разделена на подсистемы, выполняющие различные функциональные обязанности, направленные на достижение общих целей системы. Компоненты организации взаимодействуют между собой и с окружающей средой. Имеются органы управления, которые определяют цели функционирования системы, оценивают качество ее функционирования и могут изменять состав, структуру, связи и даже собственную систему управления для достижения поставленных целей (т. е. организация должна быть адаптивной и самоорганизующейся).

Целью ИСО является количественное обоснование принимаемых решений по управлению организационными системами.

Трудности решения задач организационного управления обусловлены сложностью исследуемых систем (наличие огромного количества подсистем, элементов и взаимосвязей между ними, противоречивость целей различных подсистем, необходимость учета случайных и неопределенных факторов и др.). Кроме того, в любой организационной задаче всегда фигурируют «человеческие факторы», которые трудно поддаются количественной оценке.

Под операцией будем понимать совокупность действий, мероприятий, направленных на достижение определенной цели (или группы целей), т. е. фактически любое целенаправленное действие человека является операцией. Можно сказать, что ежедневно, сами того не подозревая, мы занимаемся исследованием операций. Конечно, простейшие задачи принятия решений: брать ли с собой зонтик, отправляясь утром на работу; каким видом транспорта воспользоваться, чтобы добраться до места работы и т. п., решаются на уровне интуитивного моделирования, без использования математических методов и компьютеров. Однако, например, в задачах организационного управления производством нельзя принимать решения только на основе интуиции и здравого смысла, так как выбор неоптимального решения может привести к тяжелым экономическим последствиям.

Дисциплина «Исследование операций» получила интенсивное развитие в годы второй мировой войны, однако складываться она начала гораздо раньше. Например, некоторые простейшие модели экономических процессов были предложены Ф.Кенэ (1758 г.), А.Смитом (1776 г.), О.Курно (1838 г.), Л.Вальрасом (1874 г.). Математические основы линейного программирования успешно разрабатывались К.Жорданом (1873 г.), Г.Минковским (1896 г.), Г.Фаркашем (1903 г.). Весьма серьезные результаты получены А.А.Марковым в области динамического программирования, А.Эрлангом по теории массового обслуживания. Нашедшие широкое применение математические модели экономических процессов были разработаны в 1936 г. В.Леонтьевым (удостоен Нобелевской премии в 1973 г.), в 1937 г. Дж. фон Нейманом, в 1939 г. Л.В.Канторовичем (удостоен Нобелевской премии в 1975 г.). Большой вклад в формирование и развитие ИСО внесли также Р.Акоф, Р.Беллман, Дж.Данциг, А.Кофман, Г.Кун, Т.Саати, Р.Фор, Р.Черчмен и др.

Хотя некоторые операционные исследования были выполнены еще до первой промышленной революции, однако именно во время этой революции возник широкий круг задач организационного управления, для решения которых потребовалось развитие ИСО. По мере расширения масштабов производственного предприятия одному человеку становилось все труднее осуществлять все управленческие функции. Стали появляться должности руководителей производственного отдела, финансового отдела, отдела кадров, отдела сбыта и др. При дальнейшем развитии производства происходила еще большая дифференциация управленческих функций. Так, например, в составе производственного отдела появились подразделения, которые занимались вопросами транспортировки грузов, ремонта оборудования и транспортных средств, управления снабжением, контролем качества продукции, календарным планированием и т. п. По мере роста народонаселения и освоения новых территорий создавались новые рынки сбыта и открывались новые источники сырья. В результате происходило дальнейшее функциональное и территориальное расчленение органов управления.

Таким образом, общая задача организационного управления разбивается на ряд частных задач. Для каждого функционального подразделения должны быть определены цели и заданы критерии, позволяющие оценивать степень достижения этих целей. Трудность решения общей задачи организационного управления заключается в том, что цели различных подразделений оказываются противоречивыми, невозможно оптимизировать все критерии одновременно. Поэтому руководитель должен выбрать такую стратегию управления, которая наилучшим образом служит интересам предприятия в целом, а не интересам отдельных подразделений.

Значительные успехи были достигнуты в период 1935-1942 гг., когда в Англии, США, Канаде и Франции были сформированы специальные научные коллективы для решения задач организации военных действий. Термин «исследование операций» (имелись в виду боевые операции) впервые вошел в обиход в 1939 г. Надо отметить, что исследованию операций как новому направлению в математической науке уделялось особое внимание, поскольку оно имело огромное стратегическое значение. Первое издание книги Ф.Морза и Д.Кимбела «Методы исследования операций» (1946 г.) было засекречено, и только в 1951 г. она была опубликована в открытой печати.

Привлечение науки к решению задач организационного типа в промышленности было обусловлено второй промышленной революцией. Значительное расширение научных исследований в области связи, автоматического управления и вычислительной техники после второй мировой войны, а также выпуск электронно-вычислительных машин (ЭВМ) создали научно-техническую базу автоматизации, сущность которой заключается в передаче функций управления машине (ЭВМ). Таким образом, к началу 50-х годов ИСО получило широкое распространение в промышленности, в государственных учреждениях и учебных заведениях и получило признание как самостоятельная научная дисциплина в связи с организацией Американского общества по исследованию операций (в 1952 г.) и Международной федерации обществ по исследованию операций (в 1957 г.).

Современные промышленные предприятия и научно-производственные комплексы, научно-исследовательские и опытно-конструкторские центры и т. п. самые разнообразные по характеру своей деятельности организации производственной и непроизводственной сферы представляют собой сложные системы «человек – машина», эффективность которых существенно зависит от качества организационного управления этими системами. Чтобы добиться высокого качества управления такого рода системами, современному руководителю далеко не всегда бывает достаточно личного опыта, интуиции и организаторских способностей. При формировании управляющих решений руководитель вынужден учитывать многочисленные, нередко взаимно противоречивые соображения и опираться на сложные критерии эффективности достижения целей. При решении широкого круга задач оптимизации управляющих решений неоценимую услугу оказывает руководителю исследование операций.

В настоящее время ИСО – одна из быстро развивающихся наук, завоевывающая все более обширные области применения. Использование операционных методов в промышленности и сельском хозяйстве, в лесном и водном хозяйствах, в горнорудной и нефтегазовой отраслях экономики, в сфере образования, науки, медицины, культуры и спорта, в организациях транспорта (автомобильного, железнодорожного, воздушного и др.) и коммуникаций, в сфере торговли и коммунальных услуг, в правоохранительных и военных учреждениях, в административно-государственном аппарате является обычным явлением и охватывает широкий спектр функций управления. В зависимости от сферы применения операционные исследования затрагивают проблемы добычи полезных ископаемых, определения места размещения и производственных мощностей промышленных предприятий, выпуска промышленной продукции, распределения ресурсов, транспортировки и складирования, ремонта и замены оборудования, управления запасами, разработки производственных графиков, сетевого и календарного планирования и т. п.

ИСО как средство решения задач организационного управления можно рассматривать и как науку, и как искусство. Правомерность утверждения о научности подхода к формированию управляющих решений вытекает из того обстоятельства, что при решении возникающих проблем используются математические модели и методы. ИСО можно рассматривать и как искусство, поскольку успешное выполнение всех этапов операционного исследования во многом определяется творческими способностями и интуицией исследователей.

3.2. Методология ИСО

Дисциплина «Исследование операций» принадлежит к разряду комплексных научных дисциплин, имеющих тем не менее собственный объект изучения и собственную методологию. Отличительной особенностью ИСО является использование системного анализа и комплексный подход к решению задач организационного управления. При решении этих задач предпринимается попытка учесть все существенные факторы, установить между ними связь и оценить их с точки зрения критерия качества функционирования всей системы в целом. Задача решается с использованием комплексных научно-исследовательских коллективов, которые анализируют задачу с различных позиций (экономической, физической, биологической и т. д.). В состав операционной группы могут входить специалисты разных областей знания: математики, инженеры, экономисты, социологи, психологи и др. Целью создания подобных групп является комплексное исследование всего множества факторов, влияющих на решение проблемы, и использование идей и методов различных наук.

Другой особенностью методологии ИСО является использование моделей и применение вычислительной техники. Операционные модели имеют следующую структуру:

1) к р и т е р и й, характеризующий качество функционирования системы

$$U = f(X,Y) \to \max \text{ (или min)},$$

где X – управляемые переменные, Y – неуправляемые переменные;

2) о г р а н и ч е н и я в виде неравенств и / или равенств, определяющие допустимую область изменения управляемых переменных.

Используя методы математического анализа или численные методы решения экстремальных задач можно найти оптимальное решение, которое максимизирует или минимизирует (в зависимости от существа задачи) критерий качества U при заданных ограничениях.

Так как оптимальные значения переменных, полученные в результате решения, улучшают качество функционирования системы только в случае, когда исследуемая модель является хорошим описанием реальной системы, необходима проверка соответствия модели реальной действительности и оценка найденного решения.

И наконец, поскольку целью ИСО является улучшение качества функционирования реальных систем, результаты исследования должны быть внедрены.

Таким образом, любое операционное исследование состоит из следующих этапов: 1) постановка задачи; 2) построение модели; 3) получение решения; 4) проверка модели и оценка решения; 5) внедрение результатов.

Постановка задачи ИСО. Имеется лицо, принимающее решение (ЛПР), в распоряжении которого имеются различные стратегии (значения управляемых переменных). При выборе каждой стратегии существует возможность получения различных результатов, которые в той или иной мере соответствуют целям ЛПР. На эффективность выбираемых стратегий могут влиять условия окружающей среды (неуправляемые переменные).

При постановке задачи ИСО требуется определить круг лиц, принимающих решение (ЛПР), выделить существенные факторы, разделить их на управляемые и неуправляемые факторы, а также сформулировать цели функционирования системы.

Таким образом, приходим к необходимости рассмотрения задачи принятия решений $<\Omega, P>$, где Ω – исходное множество альтернатив (вариантов), P – принцип оптимальности, с помощью которого выделяется некоторое подмножество $\Omega_{\text{оп}} \in \Omega$ (в частности, $\Omega_{\text{оп}}$ может состоять из единственного элемента). Далее из подмножества $\Omega_{\text{оп}}$ выбирается некоторое приемлемое решение, наилучшее с точки зрения ЛПР. В общей задаче принятия решений множество вариантов Ω и принцип оптимальности P могут быть заранее не заданы. Задачу с заданным Ω называют задачей выбора, задачу с заданными Ω и P называют задачей оптимизации. В практических задачах варианты $X \in \Omega$ характеризуются совокупностью различных свойств. Предположим, что для каждого варианта X можно определить вектор характеристик $\psi(X) = (\psi_1(X), \psi_2(X), \ldots, \psi_q(X))$, где $\psi_s(X)$ является количественной оценкой варианта X по свойству s, $(s = \overline{1,q})$. С использованием этих оценок можно сформировать принцип оптимальности P.

Построение модели. Модель является представлением действительности. В ИСО модели играют исключительную роль. Модели позволяют получать ясные и компактные описания отображаемых ими систем, вскрывая механизм их работы. Анализируя модели и экспериментируя на них, обычно удается определить, как влияют изменения в рассматриваемой системе на качество ее функционирования.

Эти методы часто исключают необходимость проведения экспериментов на самой системе. Иногда проведение натурных экспериментов на системе либо вообще невозможно, либо сопряжено со слишком большими затратами финансов и времени. Обычно удается построить гораздо более простые модели, чем объекты, которые они отображают. Хотя для абсолютно точного предсказания какого-либо явления может потребоваться очень большое число переменных, то для определения его основных особенностей обычно достаточно лишь относительно небольшого числа переменных. Сложность состоит в том, чтобы выбрать наиболее существенные переменные и правильно описать связи между ними, и правильно сформулировать цель функционирования системы.

В ИСО используются модели четырех типов: 1) изобразительные, 2) аналоговые, 3) символические, 4) имитационные.

1) Изобразительные модели внешне похожи на реальный объ-ект, но отличаются от него размерами, представляя собой образы и копии этого объекта. Наиболее распространенными примерами моделей этого типа служат рисунки, чертежи, эскизы, фотографии, уменьшенные или увеличенные модели и др. В процессе конструирования и производства автомобилей, станков, мебели широко используются чертежи, в которых отображаются размеры деталей и общий вид изделия. При моделировании одежды, как правило, применяются эскизы, схемы раскроя материи. Аэродинамические свойства предлагаемой конструкции самолета предварительно исследуются на уменьшенной натурной модели в аэродинамической трубе. Для исследования гидродинамических характеристик кораблей их уменьшенные натурные модели можно испытывать в бассейнах. Всем хорошо знакома модель молекулы ДНК в виде двойной спирали, обычно используемая на уроках биологии в качестве наглядного пособия.

2) В аналоговых моделях набор одних свойств используется для отображения совершенно иных свойств, например, гидравлическую систему можно использовать в качестве аналога электрической, транспортной или экономической системы. Затухающие колебания математического маятника в вязкой среде можно моделировать с помощью электрической схемы, состоящей из конденсатора, индуктивности и резистора; в этом случае аналогом механических движений маятника является изменение напряжения на обкладках конденсатора, аналогом вязкости среды является сопротивление резистора.

3) В символических моделях описание объекта производится с помощью условных символов. Например, художественное произведение можно представить в письменном виде с помощью букв алфавита, для записи музыкального произведения могут быть использованы ноты. Если в процессе моделирования используются математические символы (знаки операций над множествами, арифметических и логических операций; равенства, неравенства, функциональные зависимости; производные и интегралы; алгебраические, дифференциальные, интегро-дифференциальные уравнения и т. п.), то получим математическую модель. Необходимо

отметить, что математические модели занимают центральное место в решении задач ИСО. Однако следует иметь в виду, что решение задач организационного управления далеко не всегда сводится к построению математических моделей и выполнению соответствующих вычислений. Это обусловлено, в частности тем, что в ходе формирования управляющих решений нередко сталкиваются с факторами, которые для правильного решения поставленной задачи являются существенными, но не поддаются строгой формализации и, следовательно, не могут непосредственно вводиться в математическую модель. Одним из трудно формализуемых факторов такого рода является фактор человеческой деятельности (или так называемые «бихевиоральные» факторы (от слова behaviour – поведение)).

4) Имитационные модели. Хотя методы математического моделирования имеют широкий диапазон применения, при рассмотрении многих важных задач организационного управления возникает необходимость обращения совершенно иным методам анализа. Дело в том, что для численного решения математических моделей приходится ограничиваться случаями, когда система обладает небольшой размерностью, вводить упрощающие предположения относительно структуры исследуемого объекта и алгоритмов его функционирования. При этом получаемые математические модели могут служить лишь грубым приближением процессов, протекающих в реальной системе. Если такое упрощение действительности недопустимо, может оказаться целесообразным использование имитационных моделей, которые позволяют «воспроизводить» (имитировать) на компьютере реакцию системы на управляющие воздействия и изменения условий окружающей среды, добиваясь при этом высокой степени адекватности поведения модели и объекта. При имитационном моделировании можно учитывать произвольные функциональные зависимости между переменными, задавая их таблично или с помощью сложных математических функций; есть возможность учета неопределенных факторов, а также генерирования случайных величин, законы распределения которых задаются таблично или имеют вид сложных математических функций; можно запрограммировать сложные алгоритмы функционирования системы. Таким образом, мы получаем возможность имитации поведения систем, для которых построение математических моделей и получение решений невозможно. Имитационные модели позволяют добиться более точного представления системы. Основной недостаток имитационного моделирования заключается в том, что требуется проведение множества экспериментов, возникают трудности с планированием эксперимента и статистической обработкой результатов наблюдений.

Принципы моделирования. Качество модели в значительной мере определяется творческими способностями операционной группы. Поэтому невозможно написать инструкцию, по которой можно было научить составлять модели конкретных задач ИСО. На основе накопленного опыта построения моделей можно выделить некоторые принципы, которые могут стимулировать воображение и помочь правильно выбрать направление творческого поиска.

1) Непосредственный анализ функционирования системы. Этот принцип используется тогда, когда структура системы достаточно проста, законы функционирования системы известны, так что ее можно описать, обследовав систему.

2) Использование аналога. В ситуации, когда структура системы достаточно очевидна, но метод ее математического описания неясен, иногда можно воспользоваться сходством рассматриваемой системы с аналогичной системой, имеющей более простую структуру.

3) Анализ данных (принцип «черного ящика»). В ситуациях, когда структура системы не очевидна, можно попытаться описать ее на основе анализа данных, характеризующих функционирование системы. Сформированную гипотезу о структуре системы необходимо апробировать, используя другие экспериментальные данные, не совпадающие с данными, на базе которых эта гипотеза построена. Многие фундаментальные законы в области физики, химии, астрономии и др. наук были установлены первоначально именно благодаря анализу экспериментальных данных и результатов наблюдений.

4) Эксперимент на системе. Когда анализ данных не позволяет выделить существенные переменные и определить влияние отдельных переменных на характеристики работы системы, иногда возникает необходимость проведения натурных экспериментов.

5) Использование «искусственной действительности.

Наиболее сложной является ситуация, когда отсутствуют описательные данные о системе или их невозможно получить, а проведение экспериментов на системе требует длительного времени и/или является дорогостоящим мероприятием, а иногда и вовсе недопустимо. Например, необходимо провести исследования с целью выяснения методов, позволяющих контролировать развитие крупных социальных конфликтов. Данные, необходимые для количественного анализа подобных ситуаций, отсутствуют и их нельзя получить. Общество, очевидно, не может также позволить проводить эксперименты в этой области. О структуре подобных ситуаций почти всегда нет никаких предположений. Как же можно приступить к построению модели конфликтов такого рода?

Прежде всего строится сложная экспериментальная ситуация, моделирующая «искусственную действительность», с помощью которой можно испытать большое число гипотез относительно структуры и свойств исследуемой системы. «Искусственная действительность» применяется не как модель действительности, а скорее представляет собой действительность, которую можно смоделировать. Она используется для набора статистики (накопления опыта). Должна существовать возможность разбиения (декомпозиции) ситуации на ряд простых экспериментальных ситуаций. Разрабатываются самостоятельные «микро»-теории для каждой элементарной ситуации (например, модели конфликтов двух лиц с использованием теории антагонистических игр) и предпринимаются попытки обобщить полученные таким образом модели в единую модель искусственной действительности. Одновременно предпринимается попытка построить теорию «искусственной действительности» с помощью непосредственного «макро»-анализа статистики, накопленной в этой искусственной реальности. Эти два направления построения моделей реализуются во взаимосвязи, пока не получают некоторую удовлетворительную модель (M_1) «искусственной действительности». Затем «искусственную действительность» модифицируют и

Рисунок 3.1. Моделирование с использованием «искусственной действительности»

предпринимают попытку обобщить полученную первоначальную модель M_1, в результате получают более общую модель M_2. Таким образом, формируют последовательность моделей $M_1, M_2, \ldots, M_q$, которые по мере усложнения приобретают все большую и большую общность (рис. 1.1).

<u>Методы упрощения действительности</u>. Перед ученым-операционистом, занимающимся построением математической модели для конкретной задачи ИСО, возникают две противоречивые цели. С одной стороны, модель должна быть полной, т. е. в ней должны быть учтены все важные факторы, от которых существенно зависит исход операции. С другой стороны, модель должна быть достаточно простой, чтобы с использованием известных математических методов можно было бы найти оптимальное решение. Учет множества мелких второстепенных факторов усложняет математический анализ модели, и если даже удается получить решения, то они с трудом поддаются осмыслению. Следовательно, при построении модели желательно упростить действительность, не утратив существенным образом точности представления.

Действительность можно упростить следующими методами:

1) Исключение переменных. Очевидно, не следует опускать переменные, оказывающее существенное влияние на систему, поэтому исключение из модели той или иной переменной должно быть основано на самом строгом анализе.

2) Изменение природы переменных (замена мало

изменяющейся переменной на константу, дискретной переменной на непрерывную, непрерывной переменной на дискретную, случайной величины на ее математическое ожидание и т. п.).

3) Изменение функциональных соотношений между переменными (линейная аппроксимация нелинейных функций, замена нелинейностей кусочно-линейными функциями, интерполяция и аппроксимация с помощью полиномов и тригонометрических функций, аппроксимация дискретных функций распределения (биномиальных, пуассоновских) непрерывными нормальными законами распределения и т. п.).

4) Изменение ограничений (добавление, исключение или модификация ограничений).

Получение решения. В самой общей форме математическая модель процесса принятия решения имеет вид

$$\text{т } U = f(X,Y) \to \max, \tag{3.2.1}$$

где X – управляемые переменные, Y – неуправляемые переменные, U – функция полезности. Кроме того, в модель могут входить ограничения, которые условно запишем в виде

$$\varphi_i(X,Y) = 0, \quad (i = \overline{1,m}). \tag{3.2.2}$$

Необходимо определить вектор управляемых переменных X, максимизирующий целевую функцию U (1.2.1), и удовлетворяющий заданным ограничениям (1.2.2). Для отыскания оптимального решения могут быть использованы различные методы математического программирования (линейного, выпуклого, нелинейного, динамического, дискретного, стохастического и др.). Очень часто классические методы решения экстремальных задач могут оказаться неприемлемыми вследствие сложности вида целевой функции U, из-за большого числа переменных и ограничений. В таких ситуациях приходится использовать итеративные методы. Сущность этого метода заключается в том, что вычислительный процесс начинают с некоторого произвольного допустимого решения из области (1.2.2), а затем применяют алгоритм, обеспечивающий улучшение значения целевой функции (1.2.1). Процесс продолжается до тех пор, пока не станет ясно, что либо дальнейшие улучшения решения будут незначительными, либо «стоимость» дальнейших вычислений достаточно высока.

Проверка модели и оценка решения. Проверка модели осуществляется с целью установления адекватности модели реальной действительности и устранения недостатков в модели. К числу этих недостатков относятся следующие:

1) Модель может включать несущественные переменные.

2) Модель может не содержать некоторых существенных переменных.

3) Возможно недостаточно точной является оценка значений некоторых существенных переменных.

4) Структура модели может оказаться неверной.

Для обнаружения перечисленных недостатков преимущественно используются статистические методы.

В большинстве задач ИСО фигурирует очень большое число переменных, влияющих на выбранный критерий оптимальности, но одни переменные оказывают весьма существенное влияние на критерий, а другие влияют на него чрезвычайно слабо. Операционист должен стремиться построить адекватную модель, включающую минимальное число переменных, выбирая из большого числа переменных наиболее существенные. Имеются различные статистические методы для определения степени влияния переменной на критерий оптимальности системы: корреляционный и регрессионный анализ, а также анализ дисперсии и ковариации.

Для оценки значения некоторой величины y проводят измерения $y_1, y_2, \dots, y_r$, на основе которых вычисляется оценка $\hat{y}$. Существуют три общих метода получения оценок по выборкам: метод моментов Пирсона, метод максимального правдоподобия Фишера и метод Байеса.

Модель может оказаться неудовлетворительной вследствие того, что она включает в себя неверные функциональные зависимости. Не существует какого-либо общего систематического метода выбора функции, оптимальной в смысле соответствия имеющимся эмпирическим данным. Выбор обычно производится на основе некоторой априорной информации о соотношениях между переменными или путем визуального анализа графического представления имеющихся данных. Чем большим разнообразием функциональных зависимостей располагает исследователь, тем лучший выбор он может сделать. Согласие функции с данными наблюдений не следует проверять, используя те же данные, которые были применены при подборе функции. Оценка адекватности модели реальной системе осуществляется использованием метода статистической проверки гипотез.

Необходимо также провести анализ устойчивости решений по отношению к изменениям параметров, ошибкам измерений и т. п.

Внедрение результатов. Целью ИСО является совершенствование показателей функционирования систем организационного управления. Эта цель может быть достигнута только при следующих условиях:

1) располагающий надлежащими полномочиями руководитель одобряет и утверждает решение, найденное в результате операционного исследования;

2) это решение правильно реализуется (внедряется);

3) эффективность решения сохраняется при всех изменениях системы и внешней среды.

Одобрение предлагаемого решения зависит от того, насколько ответственному руководителю удалось уяснить сущность этого решения.

Достижение такого понимания в свою очередь зависит от степени личного участия руководителя в процессе проведения операционного исследования. В ходе проведения исследования ученый-операционист должен индивидуально встречаться с каждым заинтересованным руководителем с целью получения консультаций и обсуждений возникающих вопросов. Нужно организовать также проведение регулярных рабочих совещаний со всеми заинтересованными руководителями.

Необходимо личное участие ученого-операциониста во внедрении полученных результатов. На этапе внедрения решения, как правило, возникают задачи, которые невозможно предвидеть на этапе проведения исследования. Поэтому обычно требуется некоторая модификация полученного решения. Операционист должен разработать детальные инструкции для тех, кому предстоит реализовывать данное решение. К разработке плана внедрения следует привлекать не только руководителей (высшее руководство и низовые звенья управления), но и всех, кого прямо или косвенно касается внедряемое решение.

В условиях, когда реализация решения задачи занимает значительный интервал времени, состояние системы, для которой было получено это решение, может измениться достаточно сильно (изменение целевой функции; изменение полезности результатов, влияющих на критерий качества функционирования системы; изменение набора управляемых переменных; изменение ограничений, наложенных на управляемые переменные; изменение значений параметров; изменение структуры системы). Системы, являющиеся объектом операционных исследований, почти всегда подвергаются существенным изменениям во времени, и поэтому методы внедрения, разработанные даже самым тщательным и оптимальным образом, с течением времени неизбежно теряют свою эффективность. Поэтому в процедурах реализации решения необходимо предусмотреть средства его коррекции.

3.3. Выбор целевой функции в задачах ИСО

Типы задач ИСО. Вид целевой функции (критерия качества функционирования системы), характеризующей выбранный принцип оптимальности P, зависит от типа задач ИСО. Различают следующие типы задач ИСО:

1) Вероятностные задачи (принятие решений в условиях риска): при выборе определенной стратегии могут быть получены различные результаты, вероятности достижения которых известны.

2) Детерминированные задачи (принятие решений в условиях полной определенности): каждая выбранная стратегия приводит к единственному результату.

3) Неопределенные задачи (точнее, задачи принятия решений в условиях неопределенностей): при выборе определенной стратегии могут быть получены различные результаты, но в отличие от вероятностных задач здесь мы не располагаем никакими статистическими характеристиками

возможных результатов. Неопределенности могут возникать из-за недостаточности информации об условиях окружающей среды (природные неопределенности, или так называемые пассивные неопределенности), а также из-за непредсказуемости поведения других лиц (противников, конкурентов, партнеров), действия которых необходимо учитывать в процессе принятия решений (это активные неопределенности, создаваемые в результате целенаправленных действий).

Выбор целевой функции. Следует отметить, что выбор целевой функции в каждом конкретном случае зависит от сути рассматриваемой задачи, поэтому ниже приводятся только общие принципы выбора критериев качества для различных типов задач ИСО.

Пусть имеются m стратегий: $C_1, C_2, \ldots, C_m$ и n результатов: $R_1, R_2, \ldots, R_n$. Введем функцию полезности $U = \| U_{ij} \|_{m \times n}$, где элемент $U_{ij} = U(R_j, C_i)$ означает полезность результата R_j при применении стратегии C_i, $(i = \overline{1,m};\ j = \overline{1,n})$.

1) В вероятностных задачах считаются известными $p(R_j \mid C_i)$ – условные вероятности получения результата R_j при применении стратегии C_i, $(i = \overline{1,m};\ j = \overline{1,n})$. В этом случае в качестве целевой функции можно выбрать ожидаемую полезность стратегии, т. е. функцию вида

$$U(C_i) = \sum_{j=1}^{n} p(R_j \mid C_i)\, U(R_j, C_i).$$

Тогда для выбора рационального поведения в условиях риска необходимо будет решить оптимизационную задачу

$$U(C_i) \to \max_{C_i}.$$

2) В детерминированных задачах при выборе стратегии C_i с вероятностью 1 получается результат R_j, поэтому в качестве целевой функции можно взять полезность стратегии, которая в данном случае имеет вид

$$U(C_i) = U(R_j, C_i)$$

и поставить задачу оптимизации для определения наилучшей стратегии:

$$U(C_i) \to \max_{C_i}.$$

3) В задачах принятия решений в условиях неопределенностей для описания выбранного принципа оптимальности могут быть использованы различные критерии рационального поведения.

Критерии рационального поведения. Приведенные ниже критерии рационального поведения в условиях неопределенностей (Лапласа, Вальда, Гурвица, Сэвиджа) обычно применяются, когда ЛПР не противостоит разумный противник, т. е. в задачах с природными (пассивными) неопределенностями. В ситуациях, когда необходимо учитывать активные неопределенности, например, поведение противника, интересы которого противоречат интересам ЛПР, для построения подходящего критерия требуется специальный подход. Вопросы выбора оптимальной стратегии поведения в таких конфликтных ситуациях относятся к теории игр.

1) Критерий Лапласа. Этот критерий опирается на известный принцип недостаточного обоснования. Поскольку вероятности получения результатов $R_1, R_2, \dots, R_n$ неизвестны, а необходимая информация для утверждения, что эти вероятности различны, отсутствуют, то можно предположить, что все эти результаты равновероятны, т. е. при условии выбора стратегии C_i имеем $p(R_j \mid C_i) = 1/n$ для всех $j = \overline{1,n}$. Таким образом, исходная задача сводится к задаче принятия решений в условиях риска (вероятностной задаче):

$$U(C_i) = \sum_{j=1}^{n} U(R_j, C_i)\, p(R_j \mid C_i) = \frac{1}{n} \sum_{j=1}^{n} U(R_j, C_i) \to \max_{C_i}.$$

2) Критерий Вальда (максиминный критерий, или критерий пессимиста). В соответствии с данным критерием ЛПР выбирает свою стратегию исходя из пессимистического предположения о том, что будет получен результат с наименьшим возможным значением полезности для каждой стратегии. При этом рациональное поведение ЛПР будет определяться из решения следующей максиминной задачи:

$$\max_{C_i} \min_{R_j} U(R_j, C_i).$$

3) Критерий Гурвица (обобщенный максимин). В этом случае в целевую функцию вносится параметр α, который можно рассматривать как показатель оптимизма:

$$\max_{C_i} [\alpha \max_{R_j} U(R_j, C_i) + (1-\alpha) \min_{R_j} U(R_j, C_i)].$$

Таким образом, в отличие от критерия Вальда, где предполагается получение наименьшей полезности $\min_{R_j} U(R_j, C_i)$, здесь возлагаются надежды на достижение полезности, лежащей между наименьшей полезностью $\min_{R_j} U(R_j, C_i)$ и наибольшей полезностью $\max_{R_j} U(R_j, C_i)$. Если $\alpha = 0$, то критерии сводится к предыдущему случаю (критерию пессимиста), а при $\alpha = 1$ критерий становится максимаксным (что характеризует крайний оптимизм). Промежуточные значения $0 < \alpha < 1$ соответствуют умеренному оптимизму.

4) Критерий Сэвиджа (критерий минимаксных потерь). Функция полезности предварительно преобразуется в функцию потерь (функцию сожаления) $W = \| W_{ij} \|_{m \times n}$, где $W_{ij} = \max\limits_{C_i} U_{ij} - U_{ij}$. Таким образом, элементы матрицы потерь представляет собой разность между тем, какую полезность получил бы ЛПР, если бы знал, какой будет результат, и тем, что он в действительности получит в каждой ситуации (т. е. возможные «сожаления» ЛПР). Далее, для определения рационального поведения ставится следующая минимаксная задача:

$$\min_{C_i} \max_{R_j} W_{ij}.$$

3.4. Многокритериальные задачи

В задачах ИСО приходится иметь дело еще одним видом неопределенностей: неопределенностью цели, которая возникает из-за того, что очень трудно сформулировать цель операционного исследования с помощью одного критерия. Например, объект можно оценить с различных точек зрения, учитывая физические (габариты, вес), экономические (затраты ресурсов, стоимость), технические (быстродействие, выполняемые функции) и другие характеристики. Таким образом, получается многокритериальная задача вида

$$\begin{aligned} &U_1 = f_1(X) \to \max, \\ &U_2 = f_2(X) \to \max, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &U_p = f_p(X) \to \max, \end{aligned}$$

(здесь все критерии записаны в виде функций, которые требуется максимизировать, поскольку минимизируемый критерий всегда может быть сведен к максимизируемому путем умножения на -1). Причем эти критерии часто оказываются противоречащими друг другу, невозможно достичь наилучших значений для всех критериев одновременно; улучшение одного критерия как правило приводит к ухудшению другого критерия.

Точка $\overline{X}$ называется *оптимальной по Парето*, если не существует X, для которой найдется какой-нибудь индекс $i \in (\overline{1, p})$ такой, что

$$\begin{aligned} &f_i(X) > f_i(\overline{X}), \\ &f_j(X) \ge f_j(\overline{X}), \quad (j = \overline{1, p};\ j \ne i), \end{aligned}$$

т. е. если невозможно улучшить хотя бы один из критериев, не ухудшая при этом какой-либо из оставшихся.

Точка X° называется *оптимальной по Джоффриону*, если она является оптимальной по Парето и, кроме того, для заданного достаточно малого числа $\delta > 0$ имеем

$$\frac{f_i(X) - f_i(X^{\circ})}{f_j(X^{\circ}) - f_j(X)} < \delta,$$

для любых X, i, j таких, что

$$i \in I = \{i \mid f_i(X) > f_i(X^{\circ})\}, \quad j \in J = \{j \mid f_j(X) < f_j(X^{\circ})\}.$$

(Хотя по i-му критерию X предпочтительнее X°, но зато по j-му критерию, наоборот, X° предпочтительнее X; имеющиеся недостатки по одним критериям несущественны по сравнению с явными преимуществами по другим критериям).

Точка $\widetilde{X}$ называется *оптимальной по Слейтеру*, если не существует X, для которой

$$f_i(X) > f_i(\widetilde{X}), \quad (i = \overline{1, p}),$$

т. е. если невозможно одновременно улучшить значения всех критериев.

Точка $\widehat{X}$ называется ε-*оптимальной*, если для заданного числа $\varepsilon > 0$ не существует X, для которой

$$f_i(X) > f_i(\widehat{X}) + \varepsilon, \quad (i = \overline{1, p}),$$

т. е. если невозможно одновременно улучшить значения всех критериев на величину больше чем ε.

Сталкиваясь с многокритериальными задачами, естественно попытаться найти способ сведения их к обычным оптимизационным задачам с одним критерием, для которых существуют хорошо разработанные методы решения. Рассмотрим наиболее употребительные способы преодоления неопределенности цели.

Метод главного критерия. Пусть среди рассматриваемых критериев $U_1, U_2, \ldots, U_p$ имеется один особо важный (главный) критерий, например, U_1. Тогда можно сформулировать задачу максимизации этого критерия

$U = U_1 \to \max$

при условиях, что значения остальных критериев должны быть не меньше, чем некоторые заданные числа: $U_2 \geq U_2^*,\ U_3 \geq U_3^*,\ \ldots,\ U_p \geq U_p^*$.

Метод свертки критериев. Вместо частных критериев $U_1, U_2, \ldots, U_p$ предлагается рассматривать один обобщенный критерий, получаемый путем аддитивной свертки:

$$U = \sum_{k=1}^{p} g_k U_k \to \max, \qquad (3.5.1)$$

где весовые коэффициенты $g_k > 0$ характеризуют степень важности соответствующего критерия $(k = \overline{1, p})$.

Обобщенный критерий может быть сформирован также путем мультипликативной свертки:

$$U = \prod_{k=1}^{p} (U_k)^{g_k} \to \max \qquad (3.5.2)$$

(здесь предполагается, что $U_k > 0 \ (k = \overline{1, p})$). Отметим, что мультипликативная свертка (3.5.2) приводится к аддитивной свертке (3.5.1), если использовать логарифмический масштаб измерения критериев.

Метод нормативных показателей. Очень часто в задачах планирования и проектирования задается некоторая система нормативов $U_1, U_2, ..., U_p$ и параметры исследуемой системы должны быть такими, чтобы удовлетворялись ограничения: $U_1 \geq U_1^*,\ U_2 \geq U_2^*,\ \dots \quad \geq U_p^*$. В таких случаях в качестве целевой функции удобно выбрать

$$U = \min_{1 \leq k \leq p} \frac{U_k}{U_k^*} \to \max \text{ (максимизация наихудшего показателя)}$$

при условиях $U_1 \geq U_1^*,\ U_2 \geq U_2^*,\ \dots\ ,\ U_p \geq U_p^*$.

Метод «идеала». Допустим, что мы решили систему однокритериальных задач $U_k \to \max$ отдельно для каждого $k = \overline{1, p}$ и нашли максимальные значения $U_1^\circ, U_2^\circ, \dots, U_p^\circ$. Точку $U^\circ = (U_1^\circ, U_2^\circ, \dots, U_p^\circ)$ в критериальном пространстве назовем «идеалом». Теперь в качестве цели можно поставить задачу минимизации отклонения от «идеала»:

$$U = \sqrt{\sum_{k=1}^{p} (U_k^\circ - U_k)^2} \to \min .$$

Метод последовательных уступок. Сначала производим упорядочивание критериев в порядке убывания их важности: $U_1 \succ U_2 \succ \ldots \succ U_p$.

Затем, решая задачу $U_1 \to \max$, определяем максимальное значение $\widetilde{U}_1$ для наиболее важного критерия.

Далее решаем задачу максимизации для следующего по важности критерия: $U_2 \to \max$ при условии $U_1 \geq \widetilde{U}_1 - \Delta U_1$, т. е. предполагая, что значение первого критерия может быть уменьшено на величину ΔU_1 ради создания благоприятных условий для максимизации второго критерия. Таким образом, определяем максимальное значение $\widetilde{U}_2$ для второго критерия.

Далее решаем задачу $U_3 \to \max$ при условиях $U_1 \geq \widetilde{U}_1 - \Delta U_1,\ U_2 \geq \widetilde{U}_2 - \Delta U_2$ (т. е. делая уступки уже по двум критериям для максимизации третьего критерия), тем самым определяем $\widetilde{U}_3$, и т. д.

Наконец, решаем задачу $U_p \to \max$ при условиях $U_1 \geq \widetilde{U}_1 - \Delta U_1$, $U_2 \geq \widetilde{U}_2 - \Delta U_2,\ \dots\ ,\ U_{p-1} \geq \widetilde{U}_{p-1} - \Delta U_{p-1}$ и определяем $\widetilde{U}_p$ – максимальное значение последнего критерия.

Метод компромиссов (В.Парето). Рассмотренные выше способы решения многокритериальных задач сводятся к составлению обобщенного критерия; исходную многокритериальную задачу выбора сводим к однокритериальной задаче и далее используем известные алгоритмы решения оптимизационных задач с одним критерием. При этом остается открытым вопрос о правомочности такой замены. К анализу многокритериальных задач можно подойти с других позиций: попытаться сократить множество сравниваемых вариантов, заранее отбраковав неконкурентоспособные варианты, которые по всем критериям уступают каким-то другим вариантам (не уменьшая при этом количество критериев, по которым производится сравнение вариантов).

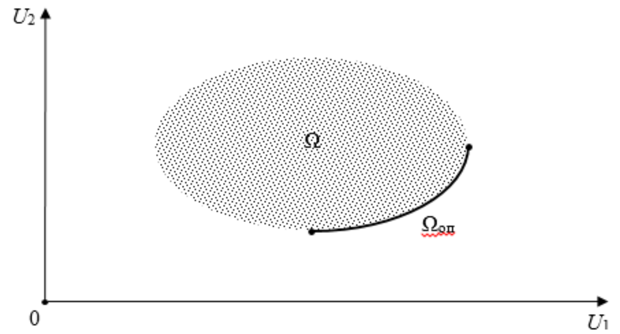


Рисунок 3.2. Парето-оптимальное множество.

В качестве примера рассмотрим задачу с двумя критериями $U_1 \to \max,\ U_2 \to \min$, где исходное множество альтернатив Ω в критериальном пространстве имеет вид, представленный на рис. 1.5.1. Производя всевозможные попарные сравнения, можно отбраковать варианты, которые не выдерживают конкуренции по сравнению с каким-либо другим вариантом, расположенным «правее» и «ниже» (для задачи $U_1 \to \max,\ U_2 \to \min$). В итоге получим множество $\Omega_{\text{оп}}$, которое является оптимальным по Парето (на Парето-оптимальном множестве невозможно улучшить значение одного критерия, не ухудшая при этом значения другого критерия). Далее ЛПР должно сделать выбор на множестве $\Omega_{\text{оп}}$. При этом приходится идти на компромиссы: чтобы улучшить значение одного критерия, надо чем-то пожертвовать (согласиться на ухудшение значения другого критерия).

Контрольные вопросы по главе 3

1. Дисциплина «Исследование операций».
2. Этапы операционного исследования.
3. Постановка задачи ИСО.
4. Типы моделей, используемых в операционном исследовании.
5. Принципы моделирования.
6. Методы упрощения действительности.

7. Типы задач ИСО.

8. Выбор целевой функции в вероятностных и детерминированных задачах.

9. Критерии рационального поведения в условиях неопределенностей:

10. Многокритериальные задачи. Оптимальность по Парето, Джоффриону и Слейтеру, ε-оптимальность.

11. Методы решения многокритериальных задач:

– метод главного критерия;
– метод свертки критериев;
– метод нормативных показателей;
– метод «идеала»;
– метод последовательных уступок;
– метод компромиссов Парето.

4. ЛИНЕЙНЫЕ МОДЕЛИ «ИССЛЕДОВАНИЕ ОПЕРАЦИЙ»

Математические модели многих практических задач относятся к типу линейных моделей, где целевая функция является линейной относительно управляемых переменных, ограничения заданы в виде линейных неравенств и / или равенств. Таким образом, линейные модели представляют собой задачи линейного программирования (ЗЛП) или целочисленного линейного программирования (ЦЗЛП). В данном разделе рассмотрены ряд содержательных постановок задач, на примере которых показано, как осуществляется постановка задачи ИСО и построение математической модели. Предполагается, что студенты уже знакомы с теорией ЛП и с теорией двойственности в ЛП, поэтому здесь рассматриваются только практические проблемы, возникающие при решении задач ИСО.

ЗЛП могут быть решены с использованием симплекс-метода, модифицированного симплекс-метода (Дж.Данциг), двойственного симплекс-метода (К.Лемке) и др.

ЦЗЛП решаются с использованием методов отсечения (Р.Гомори), метода ветвей и границ (А.Лэнд, А.Дойг), аддитивного метода (Э.Балаш) и др. Различают полностью или частично ЦЗЛП в зависимости от того, относится ли требование целочисленности ко всем переменным или к части переменных. Отметим, что первый алгоритм Гомори предназначен для решения только полностью ЦЗЛП, второй алгоритм Гомори позволяет решать полностью или частично ЦЗЛП, а с помощью третьего алгоритма Гомори решаются только полностью ЦЗЛП (в данном алгоритме в отличие от первого алгоритма исключено влияние ошибок округления на вычислительный процесс). Для некоторых классов ЗЛП и ЦЗЛП имеются специальные методы решения, учитывающие их специфику (например, для транспортной задачи, задачи выбора, задачи коммивояжера и др.).

4.1. Задача о диете (Дж.Стиглер)

Пусть имеются n видов продуктов питания, в которых содержатся m видов компонент (белки, углеводы, жиры и т. д.). Заданы c_j – стоимость единицы j-го продукта, $(j=\overline{1,n})$; α_{ij} – содержание i-й компоненты в единице j-го продукта, $(i=\overline{1,m};\ j=\overline{1,n})$; b_i – необходимое количество i-й компоненты в дневном рационе питания, $(i=\overline{1,m})$.

Введем управляемые переменные x_j – количество j-го продукта в рационе, $(j=\overline{1,n})$. Необходимо составить пищевой рацион $x=(x_1, x_2, \dots, x_n)$, имеющий минимальную стоимость при условии, что он должен обеспечивать потребности человека в различных компонентах.

Математическая модель задачи имеет вид

$$
\begin{aligned}
& z = c_1x_1 + c_2x_2 + ... + c_nx_n \to \min, \\
& \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n \ge b_1, \\
& \alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n \ge b_2, \\
& .... \\
& \alpha_{m1}x_1 + \alpha_{m2}x_2 + ... + \alpha_{mn}x_n \ge b_m, \\
& \forall x_j \ge 0.
\end{aligned}
$$

Полученная задача линейного программирования может быть решена с помощью симплекс-метода.

4.2. Задача о распределении ресурсов

Фирма производит n видов продукции, используя при этом m видов ресурсов (сырье, оборудование, трудовые ресурсы и т. д.). Заданы c_j – доход от реализации единицы j-й продукции, $(j = \overline{1,n})$; α_{ij} – затраты i-го ресурса для выпуска единицы j-й продукции, $(i = \overline{1,m};\ j = \overline{1,n})$; b_i – имеющиеся запасы i-го ресурса, $(i = \overline{1,m})$.

Введем управляемые переменные x_j – объем производства j-й продукции, $(j = \overline{1,n})$. Необходимо определить план выпуска продукций $x = (x_1, x_2, \ldots, x_n)$, который максимизирует доход фирмы при ограниченных запасах используемых ресурсов.

Составим математическую модель:

$$
\begin{aligned}
& z = c_1x_1 + c_2x_2 + ... + c_nx_n \to \max, \\
& \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n \le b_1, \\
& \alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n \le b_2, \\
& .... \\
& \alpha_{m1}x_1 + \alpha_{m2}x_2 + ... + \alpha_{mn}x_n \le b_m, \\
& \forall x_j \ge 0.
\end{aligned}
$$

Полученная модель относится к классу задачи линейного программирования и может быть решена симплекс-методом.

4.3. Межотраслевая модель экономики

Рассматривается экономическая система, состоящая из n отраслей. Предполагается, что каждая отрасль производит только один вид продукции и, наоборот, каждый вид продукции производится только одной отраслью; производственные затраты в каждой отрасли прямо пропорциональны валовому выпуску продукции в данной отрасли (чем больше выпуск, тем больше затрат).

Введем обозначения: x_i – валовой продукт i-й отрасли, $(i=\overline{1,n})$; c_i – доход, получаемый от единицы валового продукта в i-й отрасли, $(i=\overline{1,n})$; x_{ij} – затраты продукции i-й отрасли при производстве продукции в j-й отрасли, $(i,j=\overline{1,n})$; y_i – конечный продукт i-й отрасли, $(i=\overline{1,n})$.

Взаимосвязь между отраслями экономики описывается с помощью следующих соотношений (это так называемый межотраслевой баланс):

$$
\begin{aligned}
&x_1 = x_{11} + x_{12} + \ldots + x_{1n} + y_1,\\
&x_2 = x_{21} + x_{22} + \ldots + x_{2n} + y_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&x_n = x_{n1} + x_{n2} + \ldots + x_{nn} + y_n,
\end{aligned}
$$

где суммы вида $x_{i1}+x_{i2}+\ldots+x_{in}$ означают количество продукции i-й отрасли, используемой в сфере производства, $(i=\overline{1,n})$. Поскольку производственные затраты в j-й отрасли прямо пропорциональны величине валового продукта данной отрасли, то x_{ij} можно представить в виде $x_{ij}=\alpha_{ij}x_j$, $(i,j=\overline{1,n})$. Числа α_{ij}, $(i,j=\overline{1,n})$ называются коэффициентами прямых затрат («input-output» коэффициентами) и определяются по формуле $\alpha_{ij}=x_{ij}/x_j$, $(i,j=\overline{1,n})$. При этом межотраслевой баланс примет вид

$$
\begin{aligned}
&x_1 = \alpha_{11}x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1n}x_n + y_1,\\
&x_2 = \alpha_{21}x_1 + \alpha_{22}x_2 + \ldots + \alpha_{2n}x_n + y_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&x_n = \alpha_{n1}x_1 + \alpha_{n2}x_2 + \ldots + \alpha_{nn}x_n + y_n.
\end{aligned}
$$

Введем обозначения

$$
x=\begin{pmatrix} x_1\\ x_2\\ \ldots\\ x_n\end{pmatrix},\quad
y=\begin{pmatrix} y_1\\ y_2\\ \ldots\\ y_n\end{pmatrix},\quad
c=\begin{pmatrix} c_1\\ c_2\\ \ldots\\ c_n\end{pmatrix},\quad
A=\begin{pmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2n}\\ \ldots & \ldots & \ldots & \ldots\\ \alpha_{n1} & \alpha_{n2} & \ldots & \alpha_{nn}\end{pmatrix},
$$

где x – вектор валового продукта, y – вектор конечного продукта, c – вектор цен, A – матрица прямых затрат. Теперь межотраслевой баланс можно представить в матричной форме: $x - A\cdot x = y$.

Задав y – потребности общества в конечном продукте, из последнего соотношения можно определить, какой валовой продукт x обеспечивает получение заданного конечного продукта: $x=(I-A)^{-1}y$, где I – единичная $(n\times n)$-матрица. $(I-A)$ называется матрицей Леонтьева, а $(I-A)^{-1}$ – матрицей полных затрат. Как показал О.Моргенштерн, при выполнении условий

$$\alpha_{ij}\geq 0,\quad (i,j=\overline{1,n}),$$
$$\sum_{i=1}^{n}\alpha_{ij}<1,\quad (j=\overline{1,n})$$

матрица $(I-A)$ является невырожденной, т. е. существует матрица $(I-A)^{-1}$, причем она является неотрицательной (все элементы матрицы ≥ 0). В этом случае по заданному вектору конечного продукта y определяется единственный вектор валового продукта x.

Теперь для постановки оптимизационной задачи предположим, что экономика может не обеспечивать полное удовлетворение потребностей общества в конечном продукте, т. е. $x-A\cdot x\leq y$. Можно также предположить, что часть потребностей в конечном продукте удовлетворяется за счет резервов, созданных за предшествующий период: $x-A\cdot x\leq y-s$, где s – заданный неотрицательный вектор остатков. Кроме того, величина валового продукта ограничена сверху в силу ограниченности производственных потенциалов отраслей (что обусловлено ограниченностью объема основных и оборотных фондов, трудовых ресурсов и др. факторов производства): $x\leq d$.

В качестве управляемых переменных возьмем x_i, $(i=\overline{1,n})$ – компоненты вектора валового продукта. Необходимо определить x – план выпуска валового продукта, который максимизирует $z=c_1x_1+c_2x_2+...+c_nx_n$ – суммарный доход отраслей экономики с учетом балансовых соотношений, устанавливающих взаимосвязь между отраслями экономики, и ограничений на производственные потенциалы отраслей. Таким образом, получим следующую математическую модель:

$$z=c^{\mathrm{T}}x\to\max,$$
$$x-A\cdot x\leq y-s,$$
$$x\leq d,$$
$$x\geq 0,$$

которая называется статической межотраслевой моделью Леонтьева. Здесь мы также получаем задачу линейного программирования, для решения которой можно применить симплекс-метод.

4.4. Транспортная задача

Имеются m производителей и n потребителей некоторой однотипной продукции. Заданы c_{ij} – затраты на перевозку единицы продукции из i-го пункта производства в j-й пункт потребления, $(i=\overline{1,m};\ j=\overline{1,n})$; p_i – объем выпуска продукции в i-м пункте, $(i=\overline{1,m})$; q_j – спрос на продукцию в j-м пункте, $(j=\overline{1,n})$. Предположим, что выполнено условие баланса: $\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} q_j$ (суммарное количество произведенной продукции равно суммарному спросу).

Введем управляемые переменные x_{ij} – количество перевозимой продукции из i-го пункта в j-й пункт, $(i=\overline{1,m};\ j=\overline{1,n})$. Необходимо составить план перевозок $X=\| x_{ij} \|_{n\times n}$, минимизирующий суммарные транспортные затраты при условии, что вся продукция должна быть вывезена из пунктов производства и полностью должен быть удовлетворен спрос в пунктах потребления.

Математическая модель рассматриваемой задачи будет иметь вид:

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = p_i, \quad (i=\overline{1,m}),$$

$$\sum_{i=1}^{m} x_{ij} = q_j, \quad (j=\overline{1,n}),$$

$$\forall\, x_{ij} \ge 0.$$

Отметим, что выполнение условия баланса заранее предполагается, иначе система ограничений окажется несовместной и допустимое множество решений будет пустым множеством.

Полученная оптимизационная задача называется транспортной задачей и решается с использованием распределительного метода, метода потенциалов (Л.В.Канторович, М.К.Гавурин; Дж.Данциг), венгерского метода (Э.Эгервари) и др.

4.5. Задача раскроя материалов

Имеется большое количество одинаковых заготовок (рулоны, трубы, арматура, провода и т. п.) длины L. Разрезая эти заготовки, можно получить m видов деталей. Имеется n способов раскроя заготовок. Заданы l_i – длина детали i-го вида, $(i=\overline{1,m})$; α_{ij} – количество деталей i-го вида, получаемое при j-м способе раскроя, $(i=\overline{1,m};\ j=\overline{1,n})$; r_j – величина отхода при j-м способе раскроя, $(j=\overline{1,n})$; b_i – необходимое количество детали i-го вида, $(i=\overline{1,m})$.

Введем управляемые переменные x_j – количество заготовок, разрезаемых j-м способом, $(j=\overline{1,n})$. Необходимо определить оптимальное число заготовок, раскраиваемых по каждому из способов, при котором обеспечивается получение необходимого количества деталей различного вида и минимизируется суммарная величина отходов (излишнее количество полученных деталей также надо рассматривать как отходы).

Математическую модель задачи можно представить в виде:

$$z=\sum_{j=1}^{n} r_j x_j+\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\alpha_{ij}x_j-b_i\right)\cdot l_i\to\min,$$

$$\sum_{j=1}^{n}\alpha_{ij}x_j\ge b_i,\quad (i=\overline{1,m}),$$

$$\forall\, x_j\ge 0,\quad \forall\, x_j\in Z,$$

где Z – множество целых чисел. Здесь в целевой функции выражение $\left(\sum_{j=1}^{n}\alpha_{ij}x_j-b_i\right)$ означает излишнее количество полученных деталей i-го вида, $(i=\overline{1,m})$. Оптимальный план раскроя может быть найден с помощью общих методов решения ЦЗЛП (методы отсечения, метод ветвей и границ и др.).

Покажем на конкретном примере, как определяются значения α_{ij}, $(i=\overline{1,m};\ j=\overline{1,n})$ и r_j, $(j=\overline{1,n})$. Пусть имеются заготовки длины $L=70$ см, из которых можно получить $m=2$ вида деталей длины $l_1=20$ см и $l_2=15$ см. Если перечислить всевозможные способы раскроя заготовок, то получим $n=13$ способов (табл. 4.1).

Выделенная часть таблицы (матрица (2×13)) содержит значения α_{ij} – количество деталей i-го вида, получаемое при j-м способе раскроя,

$(i=1,2;\ j=\overline{1,13})$; в последней строке приведены r_j – величина отхода при j-м способе раскроя, $(j=\overline{1,13})$.

Табл. 4.1. Способы раскроя заготовок

		Способы раскроя $(j=\overline{1,13})$												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Виды деталей $(i=1,2)$	1	0	0	0	0	0	1	1	1	1	2	2	2	3
	2	0	1	2	3	4	0	1	2	3	0	1	2	0
Отходы при каждом способе раскроя		70	55	40	25	10	50	35	20	5	30	15	0	10

Аналогичным образом могут быть составлены модели для задачи раскроя материалов в двухмерном или трехмерном случаях (когда имеем дело с плоскими или объемными заготовками). Предварительный перебор всевозможных вариантов раскроя заготовок для составления матрицы $A=\|\alpha_{ij}\|_{m\times n}$ в таких задачах обычно осуществляется с помощью компьютера.

4.6. Задача о ранце

Имеются n видов продуктов питания, которыми можно заполнить ранец (рюкзак). Предполагается, что продукты упакованы в пакеты, пачки, консервные банки и т. п., их количество измеряется в целых числах. Заданы c_j – полезность (например, калорийность) единицы j-го продукта, $(j=\overline{1,n})$; α_j – вес единицы j-го продукта, $(j=\overline{1,n})$; b – максимально возможный вес ранца.

Введем управляемые переменные x_j – количество j-го продукта в ранце, $(j=\overline{1,n})$. Требуется максимизировать суммарную полезность набора продуктов $x=(x_1, x_2, \ldots, x_n)$, положенных в ранец, при условии, что вес ранца не должен превышать заданного значения.

Математическая модель задачи будет иметь вид:

$$z=c_1x_1+c_2x_2+\ldots+c_nx_n\to\max,$$
$$\alpha_1x_1+\alpha_2x_2+\ldots+\alpha_nx_n\le b,$$
$$\forall x_j\ge 0,\quad \forall x_j\in Z.$$

Отметим, что в задаче о ранце кроме условий неотрицательности и целочисленности управляемых переменных имеется только одно ограничение (на вес ранца).

Для решения данной ЦЗЛП можно воспользоваться методом динамического программирования Р.Беллмана. Этот метод предназначен для решения широких классов непрерывных и дискретных оптимизационных задач, и как оказалось, его можно успешно применить в данном случае для решения задачи о ранце.

4.7. Задача выбора (задача о назначениях)

Пусть имеются n работ и n рабочих, которые могут выполнять эти работы. Заданы c_{ij} – затраты времени на выполнение i-й работы j-м рабочим, $(i, j = \overline{1,n})$.

Введем управляемые переменные x_{ij}, $(i, j = \overline{1,n})$, имеющие следующий смысл: $x_{ij} = 1$, если для выполнения i-й работы назначен j-й рабочий, в противном случае $x_{ij} = 0$. Необходимо произвести назначение рабочих по работам с соблюдением следующих условий: для каждой работы может быть назначен только один рабочий, и наоборот, для каждого рабочего может быть назначена только одна работа. При этом требуется минимизировать суммарное количество времени, затрачиваемое на выполнение всех работ.

Математическая модель задачи выбора:

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad (i = \overline{1,n}),$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad (j = \overline{1,n}),$$

$$\forall x_{ij} \ge 0, \quad \forall x_{ij} \in Z.$$

Отметим, что в самой математической модели явно не требуется, чтобы управляемые переменные x_{ij}, $(i, j = \overline{1,n})$ принимали только два возможных значения: 0 или 1. Однако, в силу заданных ограничений получается, что они в действительности могут принимать только два значения: 0 или 1.

4.7. 1. Венгерский метод решения задачи о назначениях

Предположим, что у нас имеются 4 склада A_1, A_2, A_3, A_4 и 4 магазина B_1, B_2, B_3, B_4. Расстояние от каждого склада до каждого магазина заданы с помощью следующей матрицы

$$C=\begin{pmatrix}10 & 20 & 12 & 5\\ 3 & 14 & 9 & 1\\ 13 & 8 & 6 & 9\\ 7 & 15 & 6 & 9\end{pmatrix}$$

Требуется так прикрепить склады к магазинам, чтобы суммарное расстояние получилось минимальным. Такая задача называется задачей о назначениях. Решать ее можно с помощью так называемого венгерского алгоритма.

Алгоритм решения:

1. Решаем задачу на минимум. Цель данного шага – получение максимально возможного числа нулей в матрице С. Для этого находим в матрице С в каждой строке минимальный элемент и вычитаем его из каждого элемента соответствующей строки. Аналогично в каждом столбце вычитаем соответствующий минимальный элемент.

Если задана не квадратная матрица, то делаем её квадратной, проставляя стоимости равными максимальному числу в заданной матрице.

2. Если после выполнения первого шага можно произвести назначения, то есть в каждой строке и столбце выбрать нулевой элемент, то полученное решение будет оптимальным. Если назначения провести не удалось, то переходим к третьему шагу.

3. Минимальным числом прямых вычёркиваем все нули в матрице и среди не вычеркнутых элементов выбираем минимальный, его прибавляем к элементам, стоящим на пересечении прямых и отнимаем от всех не вычеркнутых элементов. Далее переходим к шагу 2.

Пример 4.1.

$$C=(C_{ij})=\begin{pmatrix}2 & 10 & 9 & 7\\ 15 & 4 & 14 & 8\\ 13 & 14 & 16 & 11\\ 4 & 15 & 13 & 19\end{pmatrix}.$$

Распределить ресурсы по объектам.

Решение.

1-й шаг. Значения минимальных элементов строк 1, 2, 3 и 4 равны 2, 4, 11 и 4 соответственно. Вычитая из элементов каждой строки соответствующее минимальное значение, получим

$$\begin{pmatrix} 0 & 8 & 7 & 5 \\ 11 & 0 & 10 & 4 \\ 2 & 3 & 5 & 0 \\ 0 & 11 & 9 & 15 \end{pmatrix} \longrightarrow$$

Значения минимальных элементов столбцов 1, 2, 3 и 4 равны 0, 0, 5, 0 соответственно. Вычитая из элементов каждого столбца соответствующее минимальное значение, получим

$$\longrightarrow \begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix} \longrightarrow$$

2-й шаг. Ни одно полное назначение не получено, необходимо провести модификацию матрицы стоимостей.

3-й шаг. Вычеркиваем столбец 1, строку 3, строку 2 (или столбец 2). Значение минимального невычеркнутого элемента равно 2:

$$\longrightarrow \begin{pmatrix} 0 & 8 & 2 & 5 \\ 11 & 0 & 5 & 4 \\ 2 & 3 & 0 & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix} \longrightarrow \min = 2.$$

Вычитаем его из всех невычеркнутых элементов и, складывая его со всеми элементами, расположенными на пересечении двух линий, получим

$$\longrightarrow \begin{pmatrix} 0 & 6 & 0 & 3 \\ 13 & 0 & 5 & 4 \\ 4 & 3 & 0 & 0 \\ 0 & 9 & 2 & 13 \end{pmatrix}. \quad \text{Итак,} \quad X_{\text{опт}} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Ответ. Первый ресурс направляем на 3-й объект, второй — на 2-й объект, четвертый — на 1-й объект, третий ресурс — на 4-й объект. Стоимость назначения: 9 + 4 + 11 + 4 = 28.

Примечания. 1. Если исходная матрица не является квадратной, то нужно ввести фиктивные ресурсы или фиктивные объекты, чтобы матрица стала квадратной.

4.8. Задача коммивояжера

Одна из самых известных и важных задач транспортной логистики (и класса задач оптимизации в целом) – **задача коммивояжера** (*англ. «Travelling salesman problem», TSP*). Также встречается название «*задача о* *бродячем торговце*». Суть задачи сводится к поиску оптимального, то есть кратчайшего пути проходящего через некие пункты по одному разу. Например, задача коммивояжера может применяться для нахождения самого выгодного маршрута, позволяющего коммивояжеру объехать определенные города со своим товаром по одному разу и вернуться в исходную точку. Мерой выгодности маршрута будет минимальное время, проведенное в пути, минимальные расходы на дорогу или, в простейшем случае, минимальная длина пути.

Кто и когда впервые начал исследовать задачу коммивояжера неизвестно, но одним из первых предложил решение подобной проблемы выдающийся математик XIX в. – Уильям Гамильтон. Здесь мы рассмотрим замкнутый вариант задачи (т.е. такой, когда в итоге мы возвращаемся в исходную точку) и ее решение **методом ветвей и границ**.

Методика решения задачи коммивояжера

1. Построение матриц с исходными данными

Сначала необходимо длины дорог соединяющих города представить в виде следующей таблицы:

Город	1	2	3	4
1	M	5	11	9
2	10	M	8	7
3	7	14	M	8
4	12	6	15	M

В нашем примере у нас 4 города и в таблице указано расстояние от каждого города к 3-м другим, в зависимости от направления движения (т.к. некоторые ж/д пути могут быть с односторонним движением и т.д.). Расстояние от города к этому же городу обозначено буквой M. Также используется знак бесконечности. Это сделано для того, чтобы данный отрезок путь был условно принят за бесконечно длинный. Тогда не будет смысла выбрать движение от 1-ого города к 1-му, от 2-ого ко 2-му, и т.п. в качестве отрезка маршрута.

2. Нахождение минимума по строкам

Находим минимальное значение в каждой строке (**di**) и выписываем его в отдельный столбец.

Город	1	2	3	4	di
1	M	5	11	9	**5**
2	10	M	8	7	**7**
3	7	14	M	8	**7**
4	12	6	15	M	**6**

3. Редукция строк

Производим редукцию строк – из каждого элемента в строке вычитаем соответствующее значение найденного минимума (di).

Город	1	2	3	4	di
1	M	0	6	4	5
2	3	M	1	0	7
3	0	7	M	1	7
4	6	0	9	M	6

В итоге в каждой строке будет *хотя бы одна нулевая клетка.*

4. Нахождение минимума по столбцам

Далее находим минимальные значения в каждом столбце (**dj**). Эти минимумы выписываем в отдельную строку.

Город	1	2	3	4	di
1	M	0	6	4	5
2	3	M	1	0	7
3	0	7	M	1	7
4	6	0	9	M	6
dj	0	0	1	0	

5. Редукция столбцов

Вычитаем из каждого элемента матрицы соответствующее ему dj.

Город	1	2	3	4	di
1	M	0	5	4	5
2	3	M	0	0	7
3	0	7	M	1	7
4	6	0	8	M	6
dj	0	0	1	0	

В итоге в каждом столбце будет хотя бы одна нулевая клетка.

6. Вычисление оценок нулевых клеток

Для каждой нулевой клетки получившейся преобразованной матрицы находим «**оценку**». Ею будет сумма минимального элемента по строке и минимального элемента по столбцу, в которых размещена данная нулевая клетка. Сама она при этом не учитывается. Найденные ранее di и dj не учитываются. Полученную оценку записываем рядом с нулем, в скобках.

Город	1	2	3	4
1	М	**0 (4)**	5	4
2	3	М	0	0
3	0	7	М	1
4	6	**0**	8	М

И так по всем нулевым клеткам:

Город	1	2	3	4
1	М	**0 (4)**	5	4
2	3	М	**0 (5)**	**0 (1)**
3	**0 (4)**	7	М	1
4	6	**0 (6)**	8	М

7. Редукция матрицы

Выбираем нулевую клетку с наибольшей оценкой. Заменяем ее на «**М**». Мы нашли один из отрезков пути. Выписываем его (от какого города к какому движемся, в нашем примере от 4-ого к 2-му).

Город	1	2	3	4
1	М	**0 (4)**	5	4
2	3	М	**0 (5)**	**0 (1)**
3	**0 (4)**	7	М	1
4	6	**0 (6)**	8	М

Ту строку и тот столбец, где образовалось две «М» полностью вычеркиваем. В клетку, соответствующую обратному пути, ставим еще одну букву «М» (т.к. мы уже не будем возвращаться обратно).

Город	1	2	3	4
1	М	0 (4)	5	4
2	3	М	0 (5)	**М**
3	0 (4)	7	М	1
4	6	М	8	М

8. Если полный путь еще не найден, то переходим к пункту2, если найден к пункту 9

Если мы еще не нашли все отрезки пути, то возвращаемся ко **2**-му пункту и вновь ищем минимумы по строкам и столбцам, проводим их редукцию, считаем оценки нулевых клеток и т.д.

Если все отрезки пути найдены (или найдены еще не все отрезки, но оставшаяся часть пути очевидна) – переходим к пункту **9**.

9. Вычисление итоговой длины пути маршрута

Найдя все отрезки пути, остается только соединить их между собой и рассчитать общую длину пути (стоимость поездки по этому маршруту, затраченное время и т.д.). Длины дорог соединяющих города берем из самой первой таблицы с исходными данными.
В нашем примере маршрут получился следующий: **4 → 2 → 3 → 1 → 4**.
Общая длина пути: **L = 30**.

4.8.1. Решения задачи коммивояжера венгерским методом

Исходная матрица имеет вид:

M	3	4	2	7
5	M	8	4	3
2	3	M	7	5
3	2	9	M	1
1	2	3	5	M

1. Проводим редукцию матрицы по строкам. В связи с этим во вновь полученной матрице в каждой строке будет как минимум один ноль.

M	1	2	0	5	**2**
2	M	5	1	0	**3**
0	1	M	5	3	**2**
2	1	8	M	0	**1**
0	1	2	4	M	**1**

Затем такую же операцию редукции проводим по столбцам, для чего в каждом столбце находим минимальный элемент:

M	0	0	0	5
2	M	3	1	0
0	0	M	5	3
2	0	6	M	0
0	0	0	4	M
0	**1**	**2**	**0**	**0**

После вычитания минимальных элементов получаем полностью редуцированную матрицу.

2. Методом проб и ошибок проводим поиск допустимого решения, для которого все назначения имеют нулевую стоимость.

3.

M	[-0-]	[-0-]	**[0]**	5
2	M	3	1	**[0]**
[0]	[-0-]	M	5	3
2	**[0]**	6	M	[-0-]
[-0-]	[-0-]	**[0]**	4	M

В результате получаем эквивалентную матрицу Сэ:

M	0	0	0	5
2	M	3	1	0
0	0	M	5	3
2	0	6	M	0
0	0	0	4	M

4. Методом проб и ошибок определяем матрицу назначения X, которая позволяет по аналогично расположенным элементам исходной матрицы (в квадратах) вычислить минимальную стоимость назначения.

M	[-0-]	[-0-]	**[0]**	5
2	M	3	1	**[0]**
[0]	[-0-]	M	5	3
2	**[0]**	6	M	[-0-]
[-0-]	[-0-]	**[0]**	4	M

Cmin = 2 + 3 + 2 + 2 + 3 = 12

Задача коммивояжера обычно решается с использованием методов, специально разработанных для данного класса задач (например, метод ветвей и границ для задачи коммивояжера (Дж.Литтл, К.Мурти, Д.Суини, К.Кэрел)). Отметим, что можно также воспользоваться методом динамического программирования Р.Беллмана.

Контрольные вопросы по главе 4

Сформулируйте содержательную постановку задачи, перечислите управляемые и неуправляемые факторы, а также назовите методы их решения:

– задача о диете;

– задача о распределении ресурсов;

– межотраслевая модель экономики;

– задача раскроя материалов;

– задача выбора;

– задача о ранце;

– задача коммивояжера

.

5. ТЕОРИЯ ГРАФОВ

На практике часто бывает полезно изобразить некоторую ситуацию в виде рисунков, составленных из точек (вершины) , представляющих основные ситуации, и линии (ребер), соединяющих определенные пары этих вершин и представляющих связи между ними. Таким способом удобно представлять структуру системы, в которой вершины – это блоки, а ребра – связь между блоками. Такие рисунки известны пол общим названием графов.

Графы встречаются в разных областях: структуры в гражданском строительстве, сети в электротехнике, социограммы в социологии и экономике, молекулярные структуры в химии и т.д.

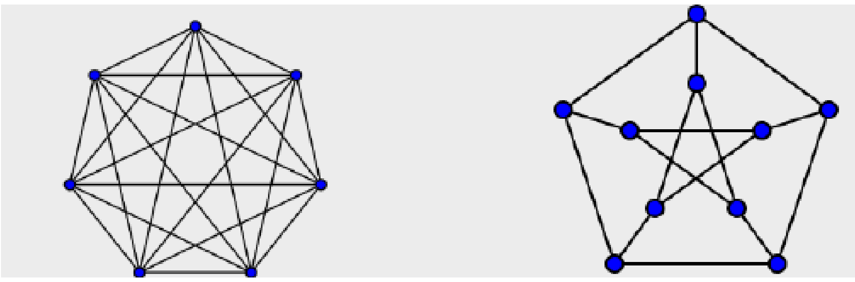
Удобны графы и при исследовании систем методом пространства состоянии. В этом случае вершины – состояние системы, процесса, ребра – действия, которые могут изменить состояние. При решении оптимизационных задач вершинами могут быть предполагаемые решения, ребрами – правила их нахождения.

Начало теории графов былао положено Эйлером в 1736 г.б когда им была написана статья о Кенигсберских мостах. Однако она была единственной в течении почти ста лет. Интерес к этой науке возродило около середины XIX в связи, с развитием естественных наук (исследования электрических сетей, моделей крист аллов, и структур молекул. Кроме того, оказалось, что многие математические головоломки могут быть сформулированы в терминах теории графа. Последние 30-40 лет ознаменовали новый период интенсивных разработок тоерии графов. Появились новые области приложения: системы телекоммуникаций, биология, психология и другие.

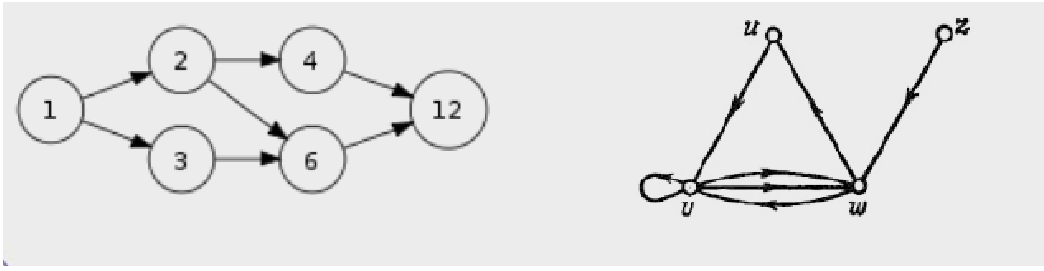
5.1. Основные понятия

Простым графом *G называется пара множеств (V,E), где V – не пустое, конечное множество элементов, называемых* ***вершинами****.* Графически это изображается точками. E – *конечное множество неупорядоченных пар различных элементов из V, называемых* ***ребрами****.* Графически это множество изображается линией, соединяющей пару точек.

Простой граф – конечный граф без петель и кратных ребер.



Ориентированным графом (орграфом) G *называется пара множеств* (V,E)*, где* V *– не пустое, конечное множество элементов (вершин).* E *– конечное множество упорядоченных пар различных элементов из* V *, называемых* ***дугами****.*



Две вершины графа, соединеные ребром, называется **смежными**.

Вершины, соединенные ребром , называется **инцидентными** этому ребру.

Два ребра, инцендентные одной вершине, называется ***смежными***.

Степенью вершины называется количество инцендентных ей ребер.

Вершине степени 0 называется ***изолированной***, степени 1 – ***висячей***.

Сумма степеней всех вершин простого графа равна удвоенному числу ребер.

В простом графе число вершин нечетной степени четно.

Маршрутом в графе называется последовательность вершин и ребер, начинающаяся и заканчивающаяся вершиной. В простом графе маршрут однозначно определяется только последовательностью вершин или ребер.

Длиной маршрута называется количество ребер в нем. Маршрут быть ***замкнутым*** и **незамкнутым**. В замкнутом маршруте первая и последняя вершины совпадает.

Маршрут в котором все ребра различны, называются ***цепью***.

Цепь, в которой нет одинаковых вершин, кроме, быть может, ее концов, называется ***простой цепью.***

Циклом называется простая замкнутая цепь. Цикл длины 1 называется ***петлей***.

Расстоянием **(**r**)** между вершинами называют длину кратчайшей цепи, соединяющей этой вершины.

Диаметром (d) называется максимальное расстояние между вершинаим в графе.

Радиус – максимальное расстояние от центра.

Способы задания графов:

1. Перечисление вершин и ребер.
2. Графическое изображение.
3. С помощью матриц смежности вершин

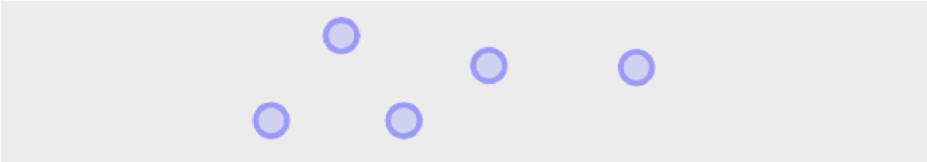
<table>
<tr><td></td><td>V_1</td><td>V_2</td><td>. . .</td><td>V_n</td></tr>
<tr><td>V_1</td><td colspan="4" rowspan="4">Количество
ребер, соединяющих
этих вершин</td></tr>
<tr><td>V_2</td></tr>
<tr><td>. . .</td></tr>
<tr><td>V_n</td></tr>
</table>

4. С помощью матриц инцинденции

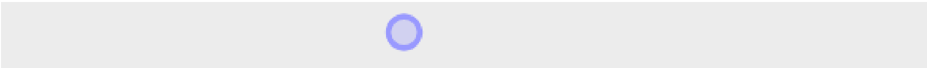
<table>
<tr><td></td><td>E_1</td><td>E_2</td><td>. . .</td><td>E_n</td></tr>
<tr><td>V_1</td><td colspan="4" rowspan="4">1,
если вершина
инцендентна
реберу</td></tr>
<tr><td>V_2</td></tr>
<tr><td>. . .</td></tr>
<tr><td>V_n</td></tr>
</table>

5.2. Основные типы графов

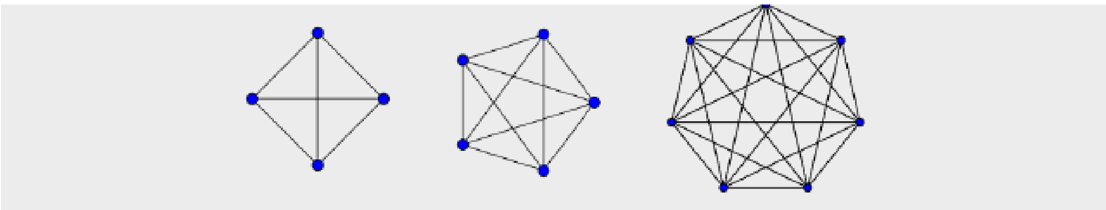
● Граф, у которого все вершины имеют степень 0, называется *пустым*, *нуль-графом* или *вполне несвязным*.



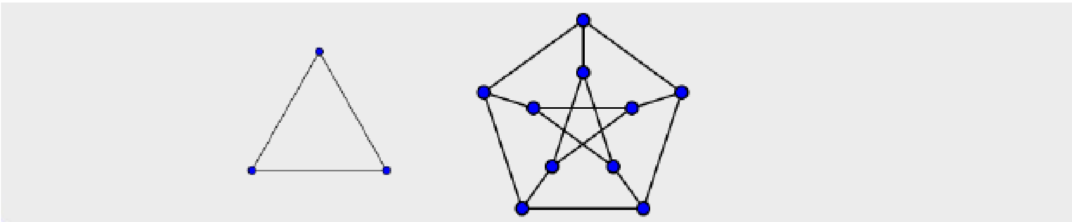
● Граф, состоящей из единственной вершины, называется *тривиальным*.



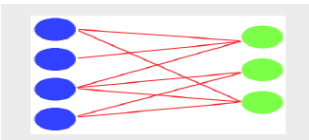
● Граф, в котором пара вершин смежна, называется *полным*.



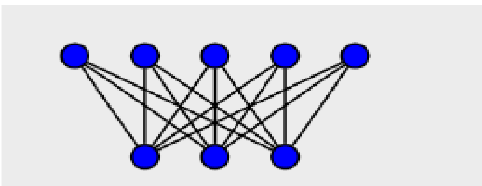
● Граф, у которого все вершины имеют одну и ту же степень r, называется *регулярным графом степени* r. Регулярный граф степени 3 называется *кубическим*.



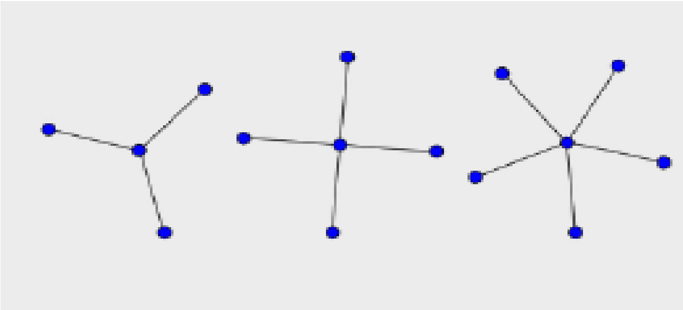
● Если множествао графа можно разделить на два не пустых и не пересекающихся подмножества таким образом, чтобы каждое ребро соединяло вершины из разных подмножеств, то такой граф называется *двухдольным*.



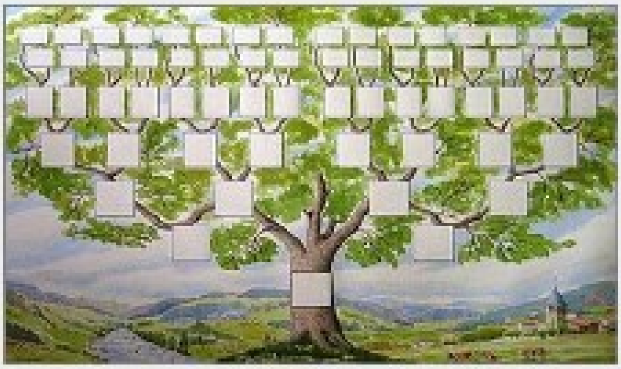
● Если при этом каждая вершина одного подмножества соединена с каждой вершиной другого, то такой граф называется *полным двудольным*



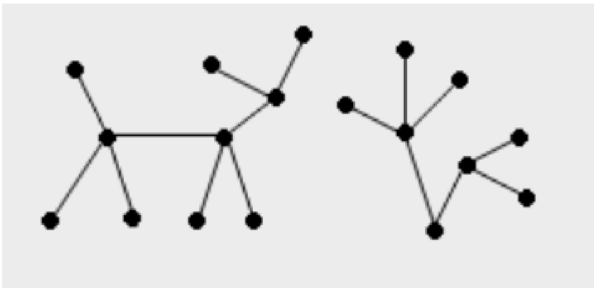
● Если в полном двудольном графе мощность одного из подмножеств равна 1, такой граф называется звездой.



● Дерево – связный граф без циклов

● Лес – граф без циклов

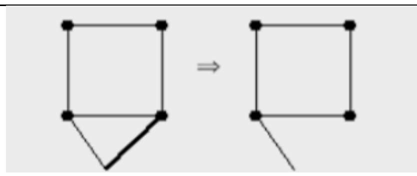
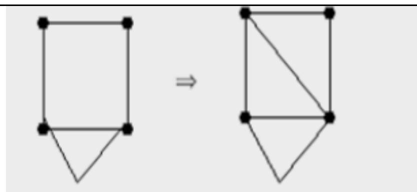
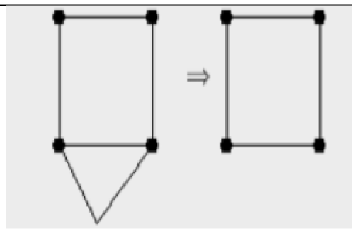
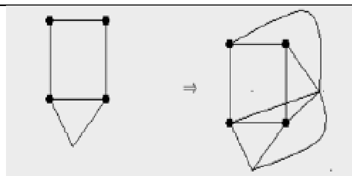
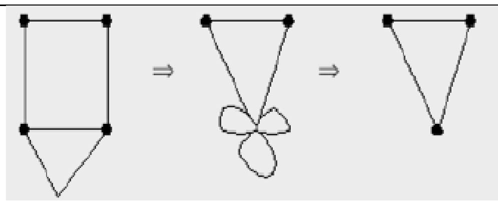


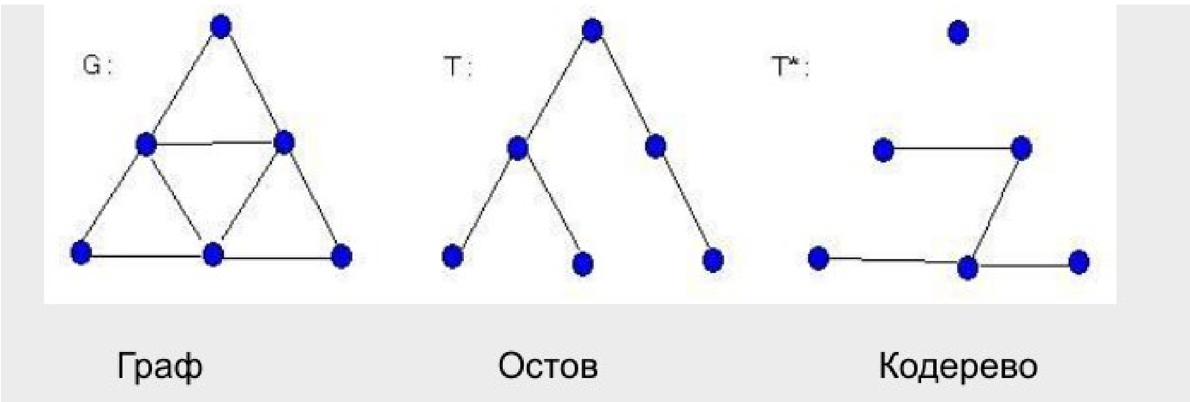
● Пусть $G(V,E)$ связный граф, n вершин, m ребер. ***Остовным деревом T*** графа $G(V,E)$ называют любой его подграф, содержащий все вершины графа $G(V,E)$ и явлющимся деревом. Остовное дерево графа $G(V,E)$ должна содержать $n-1$ ребер (ветви остова). Таким образом, любое остовное дерево есть результат удаления ровно $m-(n-1)=m-n+1$ ребер (хорды остова).

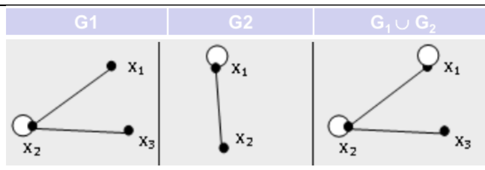
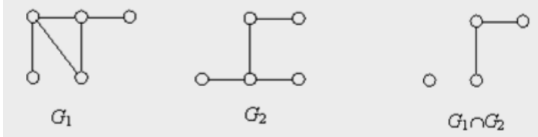
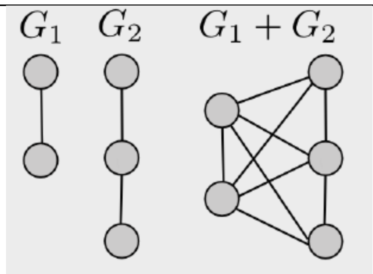
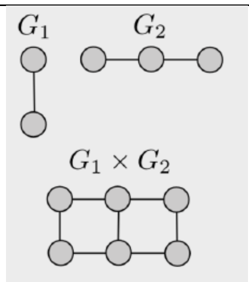
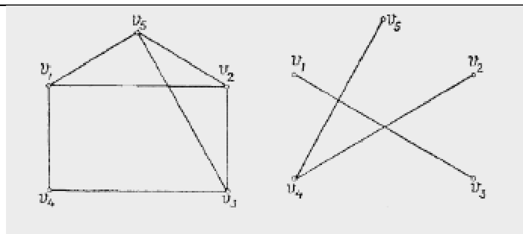
●***Кодеревом*** T^* ***остова T*** графа $G(V,E)$ называется подграф, содержащий все вершины графа $G(V,E)$ и его те ребра , которые входят в *T*.

5.3. Операции над графами

Дан граф $G(V,E)$. Построим граф $G_1(V,E_1)$.

1. Удаление ребра e. $G_1(V,E_1)$, где $E_1 = E\setminus\{e\}$	
2. Добавление ребра $G_1(V,E_1)$, где $E_1 = E\cup\{e\},\ e=(u,v)\notin E,\ u,v\in V$	
3. Удаление вершины $\{v\}$ $G_1(V,E_1)$, где $V_1 = V\setminus\{v\},\ E_1 = E\setminus\{e \mid e\ \textit{инцендентно}\ \ v\}$	
4. Добавление вершины	
5. Отождествление (слияние) вершин $u,v\in V,\ \notin V$ $G_1(V_1,E_1)$, где $V_1 =, (V\setminus\{u,v\}\cup\{w\}$, w инцендентна тем и только тем вершинам, которые были инцендентны u или v. Если u,v соединены ребром, то слияние называется стягиванием	



6. Объединение графов $G_1(V_1,E_1)$, $G_2(V_2,E_2)$ $G_1 \cup G_2 = \left(V_1 \cup V_2, E_1 \cup E_2\right)$	
7. Пересечение графов $G_1(V_1,E_1)$, $G_2(V_2,E_2)$ $G_1 \cap G_2 = \left(V_1 \cap V_2, E_1 \cap E_2\right)$	
1. Соединение графов $G_1(V_1,E_1)$, $G_2(V_2,E_2)$ $G_1 + G_2 = \left(V_1 \cup V_2, E_1 \cup E_2 \cup E_3\right)$ Множество E_3 определяется следующим образом: каждая вершина G_1 соединяется ребром с каждой вершиной G_2	
2. Произведение графов $G_1(V_1,E_1)$, $G_2(V_2,E_2)$, $G_1 \times G_2 = \left(V_1 \times V_2, E\right)$ Множество ребер E определяется следующим образом: вершины $u = (u_1, u_2)$ и $v = (v_1, v_2)$ смежны тогда и только тогда, когда $u_1 = v_1, u_2$ и v_2 смежны или u_1 и v_1 смежны, $u_2 = v_2$	
3. Дополнение графа Дополнением графа $G(V,E)$ является граф $G_1(V,E_1)$, содержащий все вершины исходного графа и толь ко те ребра, которых не хватает исходному графу для того, чтобы он стал полным. E_1: две вершины смежны тогда и только тогда, когда они не смежны в исходном графе.	

5.4. Задача о кратчайшем пути. Алгоритм Дейкстры

Задан орграф $G(V,E)$, каждой дуге (u,v) ставится в соответствие число $l(u,v)$ – длина дуги (расстояние, стоимость и т.п.). В общем случае возможно $l>0,\ l<0,\ l=0.$ Длина пути – сумма длин дуг, составляющих путь. Найти длины кратчайших путей и сами пути от фиксированной вершины s до остальных вершинт графа v_i.

Пример 5.1. Известна схема дорог. Требуется перевезти груз из одного пункта в другой по маршруту минимальной длины.

Если в графе нет циклов с отрицательной длиной, то кратчайшие пути существуют и любой кратчайший путь – это простая цепь. Наличие цикла отрицательной длины означает, что длину пути можно сделать равной $-\infty$.

Алгоритм Дейкстры

● Алгоритм Дейкстры решает задачу в случае $l \geq 0$.

● Алгоритм основан на приписывании вершинам v_i временных меток $d(v_i)$.

● Метка вершины дает верхнюю границу длины пути от s к этой вершине.

●Величины меток постопенно уменьшаются, и на каждом шаге итерации одна из временных меток становится постоянной. Это означает, что метка дает точную длину кратчайшего пути от s к рассматриваемой вершине.

Шаг 1 (начальная установка).

Положить $d(s)=0$, считать метку постоянной. $d(v_i)=\infty,\ i=\overline{1,n}$ считать метки временными. $p=s$.

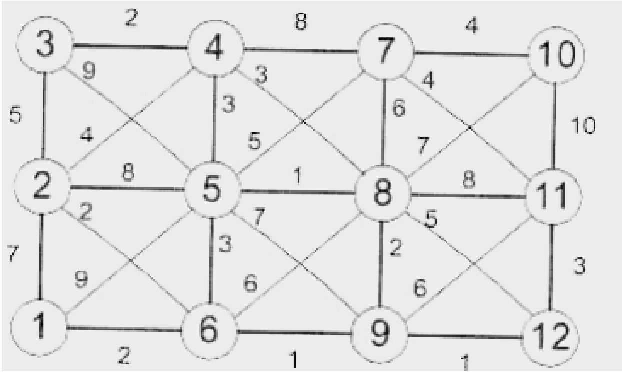
Шаг 2 (общий шаг). *Повторяется n раз, пока не будет упорядочены все вершины.*

Пересчитать временную метку $d(v_i)$ всякой неупорядоченной вершины v_i, в которую входит дуга, выходящая из p:

$$d(v_i)=\min\left(d(v_i);d(p)+l(p,v_i)\right).$$

Выбрать вершину с $\min d(v_i)$. Если их несколько, то любую. Пусть это w.

Пример 5.2. Найти кратчайшие пути от вершин $s=1$ до всех остальных вершин графа. Ребра означают две разнонаправленные дуги одинаковой длины.



Решение. Последовательность вычисления меток будем заносить в таблицу. Знаком «+» будем обозначать постоянные метки. Выполним шаг и заполним строку таблицы.

Вершины	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
Шаг 1	**0^+**	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	$p=1$

Из вершины 1 выходят дуги в вершины 2,5,6. Вычисляем метки этих вершин.

$$d(2)=\min(\infty;0+7)=7$$
$$d(5)=\min(\infty;0+9)=9$$
$$d(6)=\min(\infty;0+2)=7$$

И заполняем вторую строку таблицы

Шаг 2		**7**	∞	∞	**9**	**2^+**	∞	∞	∞	∞	∞	∞	$p=6$

Метка вершины 6 становится постоянной. Пересчитываем метки вершин, в которые можно переити из вершины 6. И так далее заполним остальные строки таблицы.

Таблица решения

Вершины	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
шаг 1	**0^+**	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	$p=1$
шаг 2		**7**	∞	∞	**9**	**2^+**	∞	∞	∞	∞	∞	∞	$p=6$
		4	∞	∞	**5**		∞	**8**	**3^+**	∞	∞	∞	$p=9$
		4^+	∞	∞	**5**		∞	**5**		∞	**9**	**4**	$p=2$
			9	**8**	**5**		∞	**5**		∞	**9**	**4^+**	$p=12$
			9	**8**	**5^+**		∞	**5**		∞	**7**		$p=5$
			9	**8**			**10**	**5^+**		∞	**7**		$p=8$
			9	**8**			**10**			**12**	**7^+**		$p=11$

			9	**8**$^{+}$			**10**			**12**			$p=4$
			9^{+}				**10**			**12**			$p=3$
							10^{+}			**12**			$p=7$
										12^{+}			$p=7$

5.5. Построение минимального остовного дерева. Алгоритм Краскала

Построение минимального остовного дерева.

Каждому ребру l графа $G(V,E)$ поставлено в соответствие число $w(l)>0$ (вес ребра). Вес дерева Т – сумма весов ребер дерева. Минмальное остовное дерево – дерево минимального веса

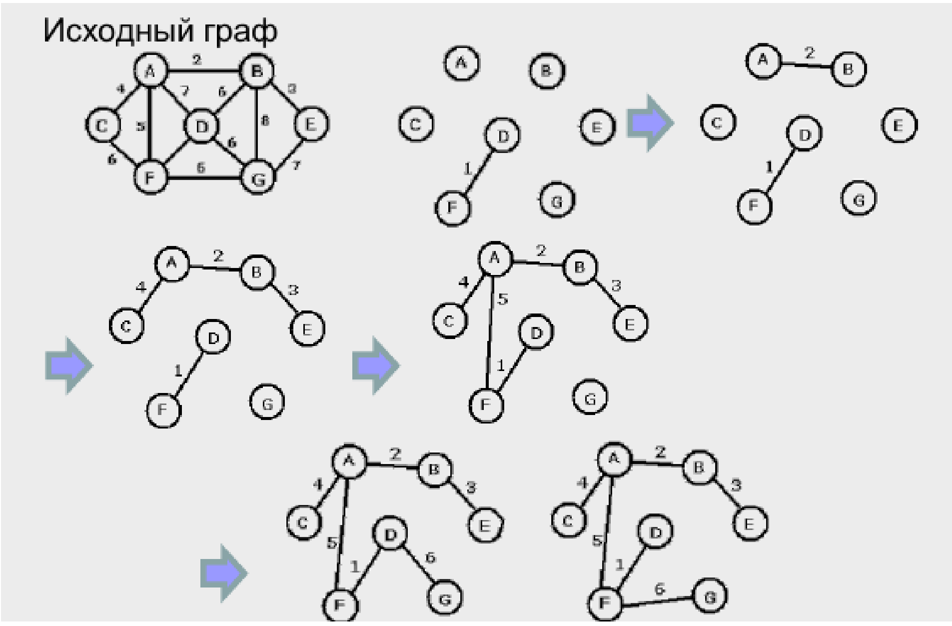
Алгоритм Краскала

Шаг 1. Упорядочить ребра в порядке возрастания весов.

Шаг 2. Включит в дерево ребро с минмальным весом.

Шаг 3. ВклюДля остальных ребр: если ребро l_i не образует цикл с уже включенными ребрами, то включить l_i в дерево, инче пропустить ребро. Закончить работу, когда будет выбраны $n-1$ ребер.

Пример5.3. Построения минимального остовного дерева



Два остовных дерева одинакового веса

Контрольные вопросы по главе 5

1. *Полный граф*
2. *Орграф*
3. *Тривиальный граф.*
4. *Матрица смежности графа*
5. *Матрица инцидентности*
6. *Маршрут в графе*
7. *Цикл в графе*
8. *Связный граф*
9. *Дерево*
10. *Лес*
11. *Остовное дерево*
12. *Кодерево остова*
13. *Двудольные графы*
14. *Алгоритм Дейкстры*
15. *Алгоритм Краскала*

6. СЕТЕВОЕ ПЛАНИРОВАНИЕ И УПРАВЛЕНИЕ

6.1. Постановка задачи сетевого планирования и управления

В современных условиях необходимо вооружить руководителей совершенным инструментом, позволяющим в любых даже самых сложных ситуациях, быстро принимать наиболее правильные решения. Поиски эффективных способов планирования сложных процессов и проектов привели к созданию методов сетевого планирования и управления (СПУ). Они применимы в тех случаях, когда конечная цель достигается путем выполнения ряда взаимоувязанных и взаимозависимых работ, входящих в единый комплекс той или иной разработки

Объектом управления в системах СПУ является коллектив, располагающий определенными ресурсами и выполняющий комплекс работ, призванный обеспечить достижение намеченной цели. Метод СПУ позволяет в любых, даже самых сложных ситуациях, быстро принимать наиболее правильные решения, выявить резервы времени и средств на одних участках работы и перебросить их на другие, более напряженные. Для отображения процесса выполнения проекта и управления им в системах СПУ используется сетевая модель.

6.1.1 История сетевого планирования и управления. Задачи сетевого планирования

Первый этап широкого использования сетевого планирования был связан с появлением диаграмм Ганта, которые появились в начале двадцатого века. Диаграмма Ганта это удобный инструмент для организации, планирования и управления ходом выполнения самых разнообразных процессов Второй этап методики сетевого планирования были разработаны в конце 50-х годов в США. В 1956 г. М. Уолкер из фирмы "Дюпон", исследуя возможности более эффективного использования принадлежащей фирме вычислительной машины Univac, объеди-нил свои усилия с Д. Келли из группы планирования капитального строительства фирмы "Ремингтон Рэнд". Они попытались использовать ЭВМ для составления планов-графиков крупных комплексов работ по модернизации заводов фирмы "Дюпон". В результате был создан рациональный и простой метод описания проекта с использованием ЭВМ. Первоначально он был назван методом Уолкера-Келли, а позже получил название метода критического пути (или CPM – Critical Path Method)]. Параллельно и независимо в военно-морских силах США был создан метод оценки и пересмотра плана PERT (Program Evaluation and Review Technique). Данный метод был разработан корпорацией "Локхид" и консалтинговой фирмой "Буз, Аллен энд Гамильтон" для реализации проекта разработки ракетной системы "Поларис", который объединял около 3800 основных подрядчиков и состоящего из 60 тыс. операций.

Использование метода PERT позволило руководству программы точно знать, что требуется делать в каждый момент времени и кто именно должен это делать, а также вероятность своевременного завершения отдельных операций.

Третий этап связан как с продолжавшимся в конце двадцатого века усовершенствованием прежних методов управления проектами, так и с появлением новых, но на более качественном уровне – с применением современного программного обеспечения и персональных компьютеров. Сначала разработка программного обеспечения велась крупными компаниями с целью поддержки собственных проектов, но вскоре первые системы управления проектами появились и на рынке программного обеспечения. Системы, стоявшие у истоков планирования,

Планирование и управление комплексом работ по проекту представляет собой сложную и, как правило, противоречивую задачу. Оценка временных и стоимостных параметров функционирования системы, осуществляемая в рамках этой задачи, производится различными методами. Среди существующих большое значение имеет метод сетевого планирования. Сетевое планирование – метод управления, который основывается на использовании математического аппарата теории графов и системного подхода для отображения и алгоритмизации комплексов взаимосвязанных работ, действий или мероприятий для достижения четко поставленной цели.

Задача сетевого планирования состоит в том, чтобы графически, наглядно и системно отобразить и оптимизировать последовательность и взаимо-зависимость работ, действий или мероприятий, обеспечивающих своевременное и планомерное достижение конечных целей. Для отображения и алгоритмизации тех или иных действий или ситуаций используются экономико-математические модели, которые принято называть сетевыми моделями, простейшие из них – сетевые графики. С помощью сетевой модели руководитель работ или операции имеет возможность системно и масштабно представлять весь ход работ или оперативных мероприятий, управлять процессом их осуществления, а также маневрировать ресурсами.

В основе сетевого планирования лежит построение сетевых диаграмм. Сетевая диаграмма (сеть, граф сети, PERT-диаграмма) – графическое отображение работ проекта и зависимостей между ними. В СПУ под термином "сеть" понимается полный комплекс работ и вех проекта с установленными между ними зависимостями. Выделяют два типа сетевых диаграмм – сетевая модель типа "вершина-работа" и "вершина-событие" или "дуги-работы". Сетевые диаграммы первого типа отображают сетевую модель в графическом виде как множество вершин, соответствующих работам, связанных линиями, представляющими взаимосвязи между работами. Так же этот тип диаграмм называют диаграммой предшествования – следования (рис 6.1).

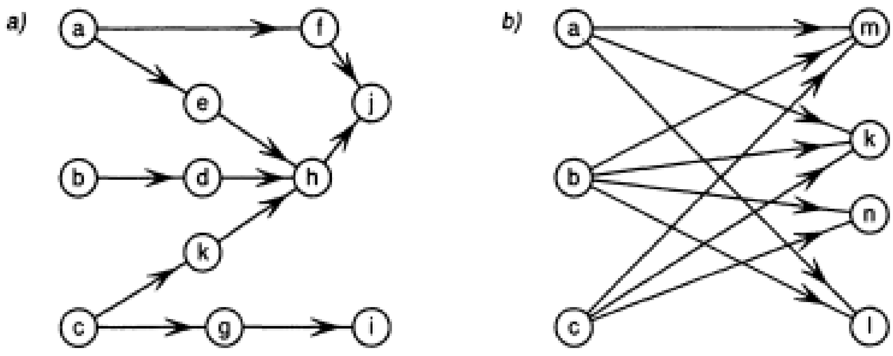


Рисунок 6.1. Фрагмент сети «вершина-работа»: *a* – нормальный пример; b – неприятный пример.

Другой тип сетевой диаграммы – сеть типа «вершина-событие». При данном подходе работа представляется в виде линии между двумя событиями (узлами графа), которые, в свою очередь, отображают начало и конец данной работы. PERT-диаграммы являются примерами этого типа диаграмм (рис 6.2).

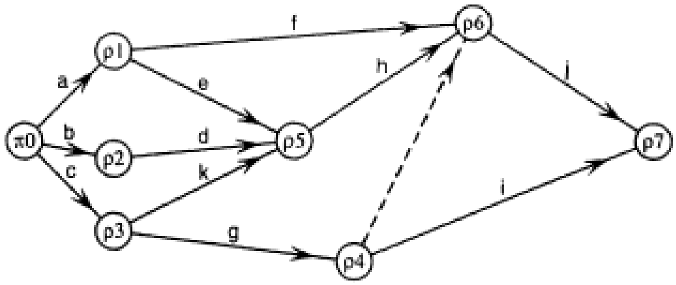


Рисунок 6.2. Фрагмент сети «вершина-событие»

6.1.2 Основные понятия сетевого планирования

Сетевое планирование и управление — это совокупность расчётных методов, организационных и контрольных мероприятий по планированию и управлению комплексом работ с помощью сетевого графика (сетевой модели).

Под *комплексом работ* мы будем понимать всякую задачу, для выполнения которой необходимо осуществить достаточно большое количество разнообразных работ.

Для того чтобы составить план работ по осуществлению больших и сложных проектов, состоящих из тысяч отдельных исследований и операций, необходимо описать его с помощью некоторой математической модели. Таким средством описания проектов является сетевая модель. *Сетевая модель* — это план выполнения некоторого комплекса взаимосвязанных работ, заданного в форме сети, графическое изображение которой называется *сетевым графиком* (рис.4.3). Главными элементами сетевой модели являются ***работы*** и ***события.***

Термин работа в СПУ имеет несколько значений. Во-первых, это ***действительная работа*** — протяжённый во времени процесс, требующий затрат ресурсов (например, сборка изделия, испытание прибора и т.п.). Каждая действительная работа должна быть конкретной, чётко описанной и иметь ответственного исполнителя. Во-вторых, это *ожидание* — протяжённый во времени процесс, не требующий затрат труда (например, процесс сушки после покраски, старения металла, твердения бетона и т.п.). В-третьих, это *зависимость*, или ***фиктивная работа*** — логическая связь между двумя или несколькими работами (событиями), не требующими затрат труда, материальных ресурсов или времени. Она указывает, что возможность одной работы непосредственно зависит от результатов другой. Естественно, что продолжительность фиктивной работы принимается равной нулю.

Событие *— это момент завершения какого-либо процесса, отражающий отдельный этап выполнения проекта*. Событие может являться частным результатом отдельной работы или суммарным результатом нескольких работ. Событие может свершиться только тогда, когда закончатся всё работы, ему предшествующие. Последующие работы могут начаться только тогда, когда событие свершится. Отсюда *двойственный характер события*: для всех непосредственно предшествующих ему работ оно является конечным, а для всех непосредственно следующих за ним — начальным. При этом предполагается, что событие не имеет продолжительности и свершается как бы мгновенно. Поэтому каждое событие, включаемое в сетевую модель, должно быть полно, точно и всесторонне определено, его формулировка должна включать в себя результат всех непосредственно предшествующих ему работ.

События на сетевом графике (или, как ещё говорят, *на графе*) изображаются кружками (вершинами графа), а работы — стрелками (ориентированными дугами):

- — событие, → — работа (процесс), - - - - - → — фиктивная

  работа — применяется для упрощения сетевых графиков (продолжительность всегда равна 0).

Существует и иной принцип построения сетей — без событий. В такой сети вершины графа означают определённые работы, а стрелки — зависимости между работами, определяющие порядок их выполнения. Сетевой график «работы–связи» в отличие от графика «события–работы» обладает известными преимуществами: не содержит фиктивных работ, имеет более простую технику построения и перестройки, включает только хорошо знакомое исполнителям понятие работы без менее привычного понятия события. Вместе с тем сети без событий оказываются значительно более громоздкими, так как событий обычно значительно меньше, чем работ (*показатель сложности сети*, равный отношению числа работ к числу событий, как правило, существенно больше единицы). Поэтому эти сети менее эффективны с точки зрения управления комплексом. Если в сетевой модели нет числовых оценок, то такая сеть называется *структурной*.

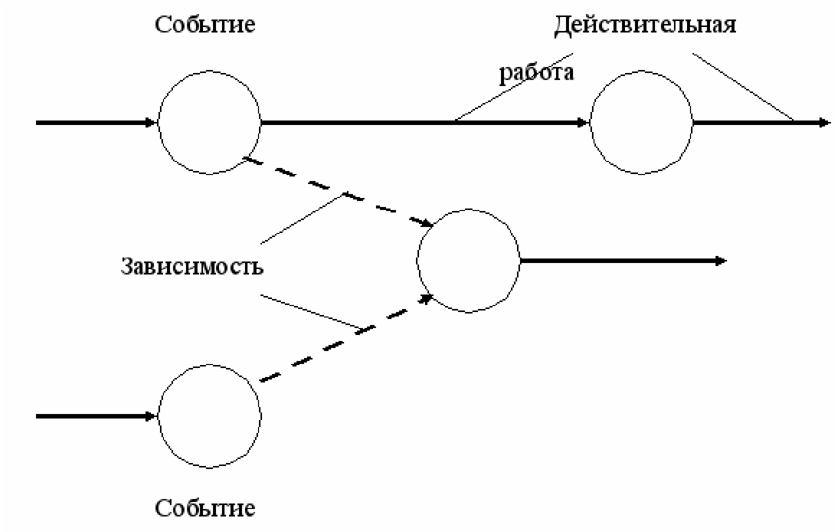


Рисунок 6.3. - Основные элементы сетевой модели

Однако на практике чаще всего используют сети, в которых заданы оценки продолжительности работ, а также оценки других параметров, например трудоёмкости, стоимости и т.п.

Событие – это факт окончания одной или нескольких работ, необходимых и достаточных для начала следующих работ. Событие не является процессом и не имеет продолжительности. Наступление события соответствует моменту начала или окончания работ (моменту формирования определенного состояния системы).

6.1.3 Порядок и правила построения сетевых графиков

Сетевые графики составляются на начальном этапе планирования. Вначале планируемый процесс разбивается на отдельные работы, составляется перечень работ и событий, продумываются их логические связи и последовательность выполнения, работы закрепляются за ответственными исполнителями. С их помощью и с помощью нормативов, если таковые существуют, оценивается продолжительность каждой работы. Затем составляется (*сшивается*) сетевой график. После упорядочения сетевого графика рассчитываются параметры событий и работ, определяются резервы времени и *критический путь*. Наконец, проводятся анализ и оптимизация сетевого графика, который при необходимости вычерчивается заново с пересчётом параметров событий и работ.При построении сетевого графика необходимо соблюдать ряд правил:

1. *Сетевые модели следует строить от начала к окончанию, т.е. слева направо;*
2. *В сетевом графике стрелки, обозначающие работы, ожидания или зависимости, могут иметь различный наклон и длину*, но должны идти слева направо, не отклоняясь влево от оси ординат, и всегда направляться от предшествующего события к последующему, т.е. от события с меньшим порядковым номером к событию с большим порядковым номером;
3. *При построении сетевого графика можно начинать последующую работу, не ожидая полного завершения предшествующей*. В этом случае нужно "расчленить" предшествующую работу на две, введя дополнительное событие в том месте предшествующей работы, где может начаться новая;
4. *В сетевой модели не должно быть «тупиковых» событий, то есть событий, из которых не выходит ни одна работа, за исключением завершающего события*. Здесь либо работа не нужна и её необходимо аннулировать, либо не замечена необходимость определённой работы, следующей за событием для свершения какого-либо последующего события. В таких случаях необходимо тщательное изучение взаимосвязей событий и работ для исправления возникшего недоразумения.
5. *В сетевом графике не должно быть «хвостовых» событий (кроме исходного), которым не предшествует хотя бы одна работа*. Обнаружив в сети такие события, необходимо определить исполнителей предшествующих им работ и включить эти работы в сеть.
6. *В сети не должно быть замкнутых контуров и петель, то есть путей, соединяющих некоторые события с ними же самими*. При возникновении контура (а в сложных сетях, то есть в сетях с высоким показателем сложности, это встречается довольно часто и обнаруживается лишь при помощи ЭВМ) необходимо вернуться к исходным данным и путём пересмотра состава работ добиться его устранения.
7. *Любые два события должны быть непосредственно связаны не более чем одной работой-стрелкой*. Нарушение этого условия происходит при изображении параллельно выполняемых работ. Если эти работы так и оставить, то произойдёт путаница из-за того, что две различные работы будут иметь одно и то же обозначение. Однако содержание этих работ, состав привлекаемых исполнителей и количество затрачиваемых на работы ресурсов могут существенно

отличаться. В этом случае рекомендуется ввести *фиктивное событие* и *фиктивную работу*, при этом одна из параллельных работ замыкается на это фиктивное событие. Фиктивные работы изображаются на графике пунктирными линиями (рис.4.4).

8. *В сети рекомендуется иметь одно исходное и одно завершающее событие*. Если в составленной сети это не так, то добиться желаемого можно путём введения фиктивных событий и работ.

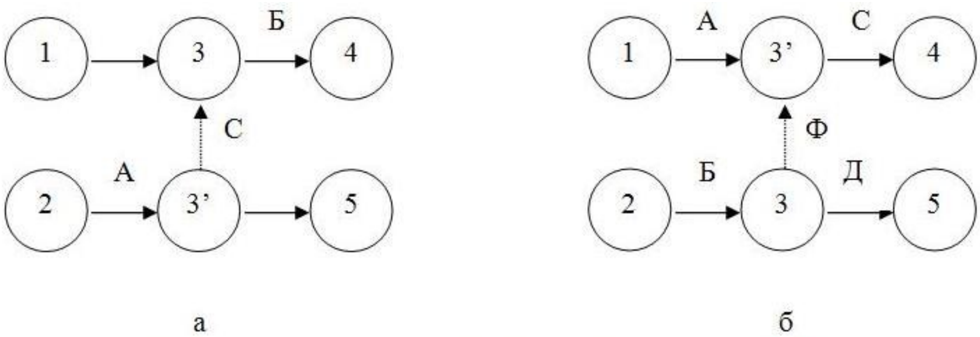


Рисунок 6.4 - Примеры введения фиктивных событий

Фиктивные работы и события необходимо вводить в ряде других случаев. Один из них — отражение зависимости событий, не связанных с реальными работами. Например, работы А и Б (рис. 4.4 а) могут выполняться независимо друг от друга, но по условиям производства работа Б не может начаться раньше, чем окончится работа А. Это обстоятельство требует введения фиктивной работы С.

Другой случай — неполная зависимость работ. Например работа С требует для своего начала завершения работ А и Б, на работа Д связана только с работой Б, а от работы А не зависит. Тогда требуется введение фиктивной работы Ф и фиктивного события 3’, как показано на рис. 4.4, б. Кроме того, фиктивные работы могут вводиться для отражения реальных отсрочек и ожидания. В отличие от предыдущих случаев здесь фиктивная работа характеризуется протяжённостью во времени.

Если сеть имеет одну конечную цель, то программа называется одноцелевой. Сетевой график, имеющий несколько завершающих событий, называется многоцелевым и расчет ведется относительно каждой конечной цели. Примером может быть строительство жилого микрорайона, где ввод каждого дома является конечным результатом, и в графике по возведению каждого дома определяется свой критический путь.

Упорядочение сетевого графика заключается в таком расположении событий и работ, при котором для любой работы предшествующее ей событие расположено левее и имеет меньший номер по сравнению с завершающим эту работу событием. Другими словами, в упорядоченном сетевом графике все работы-дуги направлены слева направо: от событий с меньшими номерами к событиям с большими номерами.

Критическим называется наиболее продолжительный из полных путей. Критический путь определяет минимально необходимое время выполнения всех работ, называемое критическим сроком. Работы и события, лежащие на критическом пути, называются критическими.

6.1.4 Упорядочение сетевого графика

Предположим, что при составлении некоторого проекта выделено 12 событий: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 и 24 связывающие их работы: (0, 1), (0, 2), (0, 3), (1, 2), (1, 4), (1, 5), (2, 3), (2, 5), (2, 7), (3, 6), (3, 7), (3, 10), (4, 8), (5, 8), (5, 7), (6, 10), (7, 6), (7, 8), (7, 9), (7, 10), (8, 9), (9, 11), (10, 9), (10, 11).

Составили исходный сетевой график 1 (рис. 6.5).

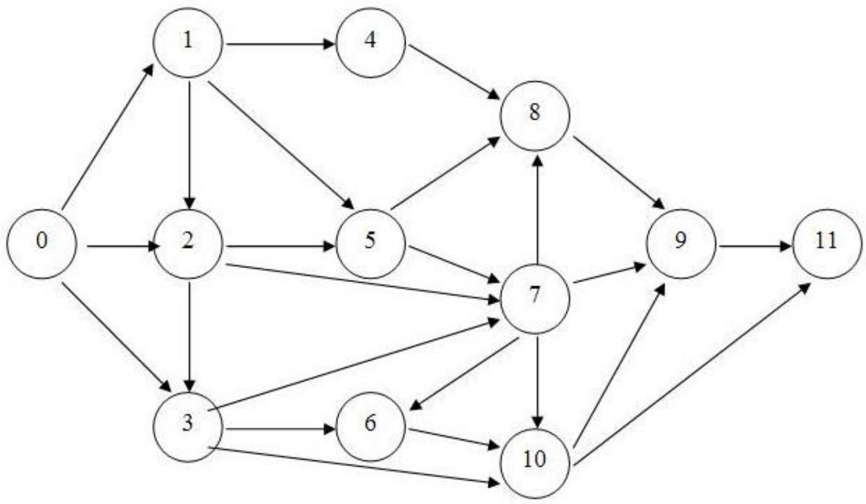


Рисунок 6.5. - Неупорядоченный сетевой график

Упорядочение сетевого графика заключается в таком расположении событий и работ, при котором для любой работы предшествующее ей событие расположено левее и имеет меньший номер по сравнению с завершающим эту работу событием. Другими словами, в упорядоченном сетевом графике все работы-стрелки направлены слева направо: от событий с меньшими номерами к событиям с большими номерами. Разобьём исходный сетевой график на несколько вертикальных слоёв (обводим их пунктирными линиями и обозначаем римскими цифрами). Поместив в I слое начальное событие 0, мысленно вычеркнем из графика это событие и все выходящие из него работы-стрелки. Тогда без входящих стрелок останется событие 1, образующее II слой. Вычеркнув мысленно событие 1 и все выходящие из него работы, увидим, что без входящих стрелок остаются события 4 и 2, которые образуют III слой. Продолжая этот процесс, получим сетевой график 2 (рис. 4.6).

Упорядочение сетевого графика заключается в таком расположении событий и работ, при котором для любой работы предшествующее ей событие расположено левее и имеет меньший номер по сравнению с завершающим эту работу событием. Другими словами, в упорядоченном сетевом графике все работы-стрелки направлены слева направо: от событий с меньшими номерами к событиям с большими номерами. Разобьём исходный сетевой график на несколько вертикальных слоёв (обводим их пунктирными линиями и обозначаем римскими цифрами). Поместив в I слое начальное событие 0, мысленно вычеркнем из графика это событие и все выходящие из него работы-стрелки. Тогда без входящих стрелок останется событие 1, образующее II слой. Вычеркнув мысленно событие 1 и все выходящие из него работы, увидим, что без входящих стрелок остаются события 4 и 2, которые образуют III слой. Продолжая этот процесс, получим сетевой график 2 (рис. 4.6).

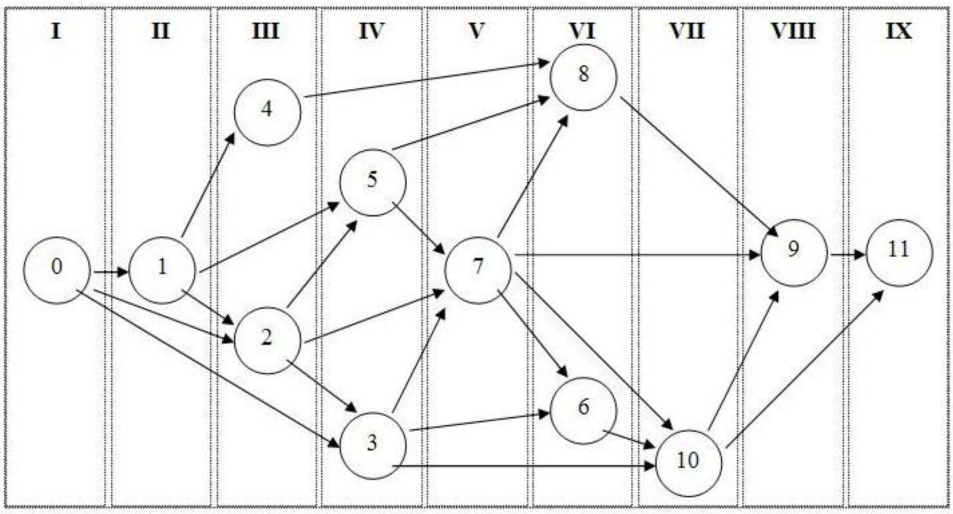


Рисунок 6.6. - Упорядочение сетевого графика с помощью слоёв

Теперь видим, что первоначальная нумерация событий не совсем правильная: так, событие 6 лежит в VI слое и имеет номер, меньший, чем событие 7 из предыдущего слоя. То же можно сказать о событиях 9 и 10.

Изменим нумерацию событий в соответствии с их расположением на графике и получим упорядоченный сетевой график 3 (рис.4.8). Следует заметить, что нумерация событий, расположенных в одном вертикальном слое, принципиального значения не имеет, так что нумерация одного и того же сетевого графика может быть неоднозначной.

Путь – это любая последовательность работ в сети, в которой конечное событие каждой работы этой последовательности совпадает с начальным событием следующей за ней работы. Путь от исходного до завершающего события называется полным. Путь от исходного до данного промежуточного события называется путем, предшествующим этому событию. Путь, соединяющий какие-либо два события, из которых ни одно

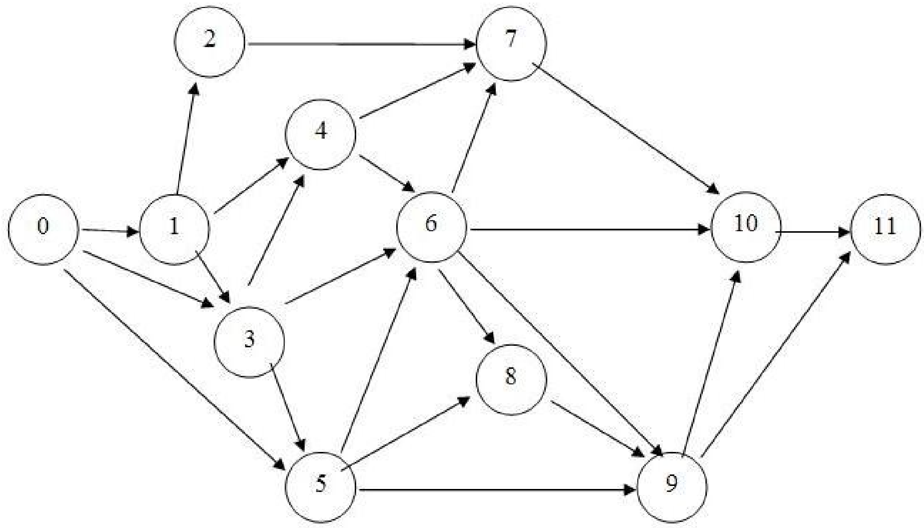


Рисунок 6.7. - Упорядоченный сетевой график

не является исходным или завершающим, называется путем между этими событиями. Продолжительность пути определяется суммой продолжительностей составляющих его работ. Путь, имеющий максимальную длину, называют *критическим*. Критическими называются также работы и события, находящиеся на этом пути

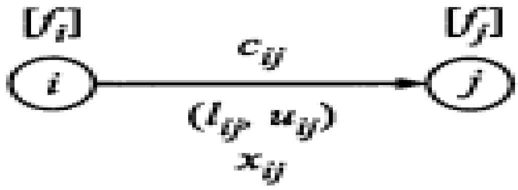


Рисунок 6.8. Условные обозначения в сетевом графике: x_{ij} – величина потока, протекающего от узла i к узлу j; u_{ij} (l_{ij}) – верхняя (нижняя) граница пропускной способности дуги (i, j); c_{ij} – стоимость прохождение единицы потока по дуге (i, j); f_i – величина чистого результирующего потока, протекающего через узел i.

Различают различные типы связей в сетевой модели:

- начальные работы;
- конечные работы;
- последовательные работы;
- работы (операции) дробления;
- работы (операции) слияния;
- параллельные работы.

При составлении сетевых графиков (моделей) используют условные обозначения (рис 4.8).

С математической точки зрения, сетевой график представляет собой связанный ориентированный граф без петель и контуров. Вершина, в которую дуги только входят, но не выходят, называется стоком. Любой путь в сетевом графике от истока к стоку называется полным. Если дугам (ребрам) графа сопоставлены какие-то числовые характеристики – весами. Вершина x_i предшествует в графе вершине x_j, если существует путь из x_i в x_j.

Различают различные типы связей в сетевой модели:

- начальные работы;
- конечные работы;
- последовательные работы;
- работы (операции) дробления;
- работы (операции) слияния;
- параллельные работы.

При составлении сетевых графиков (моделей) используют условные обозначения (рис 4.8).

С математической точки зрения, сетевой график представляет собой связанный ориентированный граф без петель и контуров. Вершина, в которую дуги только входят, но не выходят, называется стоком. Любой путь в сетевом графике от истока к стоку называется полным. Если дугам (ребрам) графа сопоставлены какие-то числовые характеристики – весами. Вершина x_i предшествует в графе вершине x_j, если существует путь из x_i в x_j.

Граф называется *связанным,* если любые его две вершины можно соединить путем, в котором не учитывается ориентация дуг.

Сетевой график – это связанный взвешенный орграф без контуров (петель).

На изображении комплекса работ с помощью сетевого графика основано сетевое планирование и управление (СПУ).

На сетевом графике 4 (рис. 6.9) критический путь проходит через работы (1;2), (2;5), (5;6), (6;8) и равен 16. Это означает, что все работы будут закончены за 16 единиц времени. Критический путь имеет особое значение в системе СПУ, так как работы этого пути определят общий цикл завершения всего комплекса работ, планируемых при помощи сетевого графика. Зная дату начала работ и продолжительность критического пути, можно установить дату окончания всей программы. Любое увеличение продолжительности работ, находящихся на критическом пути, задержит выполнение программы. На стадии управления и контроля над ходом выполнения программы основное внимание уделяется работам, находящимся на критическом пути или в силу отставания попавшим на критический путь.

Для сокращения продолжительности проекта необходимо в первую очередь сокращать продолжительность работ, лежащих на критическом пути.

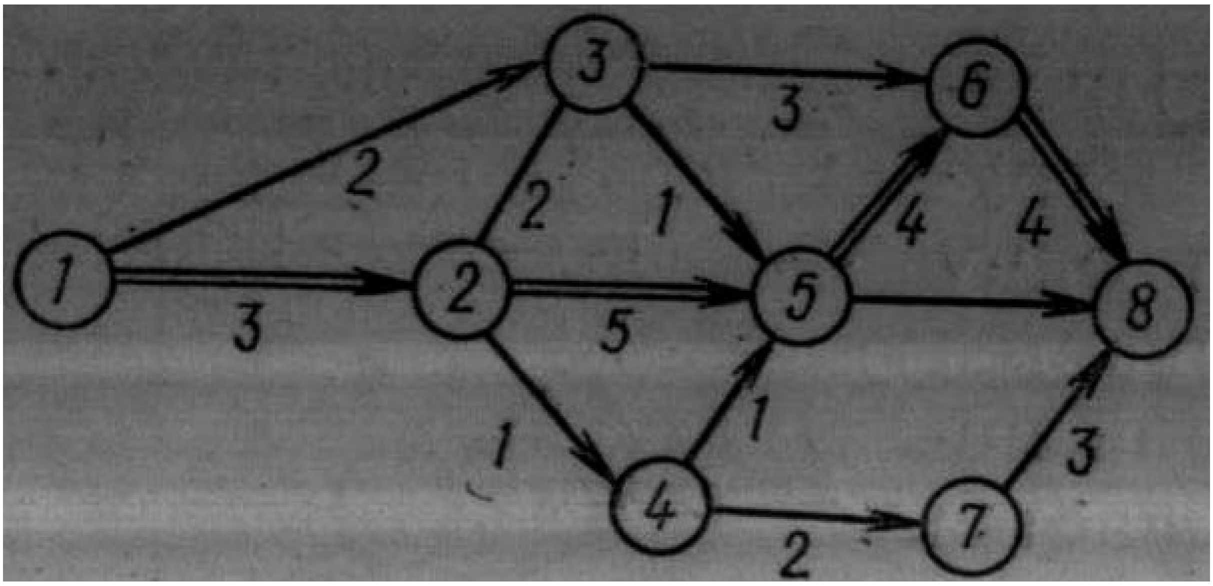


Рисунок 6.9 - Критический путь

6.1.5 Временные параметры сетевых графиков

Ранний (или ожидаемый) срок свершения события определяется продолжительностью максимального пути, предшествующего этому событию.

Задержка свершения события по отношению к своему раннему сроку не отразится на сроке свершения завершающего события (а значит, и на сроке выполнения комплекса работ) то тех пор, пока сумма срока свершения этого события и продолжительности (длины) максимального из последующих за ним путей не превысит длины критического пути.

Поэтому *поздний (или предельный) срок свершения события* равен разности максимального времени наступления последующего за работой события и времени работы до этого (будущего) события.

Резерв времени события определяется как разность между поздним и ранним сроками его свершения.

Резерв времени события показывает, на какой допустимый период времени можно задержать наступление этого события, не вызывая при этом увеличения срока выполнения комплекса работ.

Критические события резервов времени не имею, так как любая задержка в свершении события, лежащего на критическом пути, вызовет такую же задержку в свершении завершающего события.

Из этого следует, что для того, чтобы определить длину и топологию критического пути, вовсе не обязательно перебирать все полные пути

сетевого графика и определять их длины. Определив ранний срок наступления завершающего события сети, мы тем самым определяем длину критического пути, а, выявив события с нулевыми резервами времени, определяем его топологию.
Если сетевой график имеет единственный критический путь, то этот путь проходит через все критические события, то есть события с нулевыми резервами времени. Если критических путей несколько, то выявление их с помощью критических событий может быть затруднено, так как через часть критических событий могут проходить как критические, так и некритические пути. В этом случае для определения критических путей рекомендуется использовать *критические работы*.

Отдельная работа может начаться (и окончиться) в ранние, поздние или другие промежуточные сроки. В дальнейшем при оптимизации графика возможно любое размещение работы в заданном интервале, называемом *продолжительностью работы*.

Очевидно, что *ранний срок начала работы* совпадает с ранним сроком наступления предшествующего события. *Ранний срок окончания работы* совпадает с ранним сроком свершения последующего события. *Поздний срок начала работы* совпадает с поздним сроком наступления предшествующего события. *Поздний срок окончания работы* совпадает с поздним сроком наступления последующего события.

Таким образом, в рамках сетевой модели моменты начала и окончания работы тесно связаны с соседними событиями соответствующими ограничениями.
Если путь не критический, то он имеет *резерв времени*, определяемый как разность между длиной критического пути и рассматриваемого. Он показывает, на сколько в сумме могут быть увеличены продолжительности всех работ, принадлежащих этому пути. Отсюда можно сделать вывод, что любая из работ пути на его участке, не совпадающем с критическим путём (замкнутым между двумя событиями критического пути), обладает резервом времени.

Среди резервов времени работ выделяют четыре разновидности. *Полный резерв времени* работы показывает, на сколько можно увеличить время выполнения данной работы при условии, что срок выполнения комплекса работ не изменится.
Полный резерв времени работы равен резерву максимального из путей, проходящего через данную работу. Этим резервом можно располагать при выполнении данной работы, если её начальное событие свершится в самый ранний срок, и можно допустить свершение конечного события в его самый поздний срок.

Важным свойством полного резерва времени работы является то, что он принадлежит не только этой работе, но и всем полным путям, проходящим через неё. При использовании полного резерва времени только для одной работы резервы времени остальных работ, лежащих на максимальном пути, проходящем через неё, будут полностью исчерпаны. Резервы времени работ, лежащих на других (немаксимальных по длительности) путях, проходящих через эту работу, сократятся соответственно на величину использованного резерва.

Остальные резервы времени работы являются частями её полного резерва.

Частный резерв времени первого вида есть часть полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом позднего срока её начального события. Этим резервом можно располагать при выполнении данной работы в предположении, что её начальное и конечное события свершаются в свои самые поздние сроки

Частный резерв времени второго вида, или *свободный резерв времени* работы представляет часть полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом раннего срока её конечного события. Этим резервом можно располагать при выполнении данной работы в предположении, что её начальное и конечное события свершатся в свои самые ранние сроки.

Свободным резервом времени можно пользоваться для предотвращения случайностей, которые могут возникнуть в ходе выполнения работ. Если планировать выполнение работ по ранним срокам их начала и окончания, то всегда будет возможность при необходимости перейти на поздние сроки начала и окончания работ.

Независимый резерв времени работы — часть полного резерва времени, получаемая для случая, когда все предшествующие работы заканчиваются в поздние сроки, а все последующие работы начинаются в ранние сроки.

Использование независимого резерва времени не влияет на величину резервов времени других работ. Независимые резервы стремятся использовать тогда, когда окончание предыдущей работы произошло в поздний допустимый срок, а последующие работы хотят выполнить в ранние сроки. Если величина независимого резерва равна нулю или положительна, то такая возможность есть. Если же эта величина отрицательна, то этой возможности нет, так как предыдущая работа ещё не оканчивается, а последующая уже должна начаться. То есть отрицательное значение этой величины не имеет реального смысла. Фактически независимый резерв имеют лишь те работы, которые не лежат на максимальных путях, проходящих через их начальные и конечные события.

Таким образом, если частный резерв времени первого вида может быть использован на увеличение продолжительности данной и последующих работ без затрат резерва времени предшествующих работ, а свободный резерв времена — на увеличение продолжительности данной и предшествующих работ без нарушения резерва времени последующих работ без нарушения резерва времени последующих работ, то независимый резерв времени может быть использован для увеличения продолжительности только данной работы.

Работы, лежащие на критическом пути, так же как и критические события, резервов времени не имеют.

Рисунок 6.10 - Ключ к расчёту секторным методом

Следует отметить, что в случае достаточно простых сетевых графиков кроме табличного метода расчета параметров сетевых графиков, может быть применено *секторное представление* временных параметров, то есть расчет параметров может быть произведен на самом графике. Каждое событие для этого делится на четыре сектора. В левом секторе события записывают раннее начало работы, в правом — позднее окончание, в верхнем — номер данного события, в нижнем — номер предшествующего события, из которого к данному событию идёт путь максимальной продолжительности. Имеет место, когда в нижнем секторе ставят номер события и верхний сектор не заполняют. Определённые резервы времени записывают под стрелкой в виде дроби: в числителе общий резерв, а в знаменателе частный резерв.

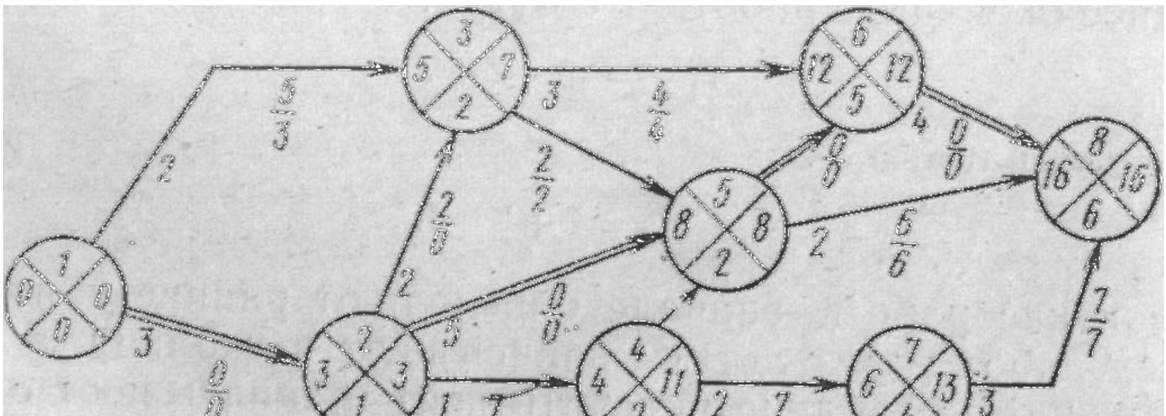


Рисунок 6.11 - Секторное представление временных параметров

Реально на практике продолжительность работ, фактическое их состояние могут изменяться. При этом может изменяться и ожидаемое время наступления события, окончания работ и критический путь. Зная критический путь, руководство может сосредоточиться на тех работах, которые являются решающими с точки зрения сроков окончания всех работ.

6.1.6 Анализ и оптимизация сетевого графика

После нахождения критического пути и резервов времени работ и оценки вероятности выполнения проекта в заданный срок должен быть проведён всесторонний анализ сетевого графика и приняты меры по его оптимизации. Этот весьма важный этап в разработке сетевых графиков раскрывает основную идею СПУ. Он заключается в приведении сетевого графика в соответствие с заданными сроками и возможностями организации, разрабатывающей проект.

Оптимизация сетевого графика в зависимости от полноты решаемых задач может быть условно разделена на частную и комплексную. Видами *частной оптимизации* сетевого графика являются: минимизация времени выполнения комплекса работ при заданной его стоимости; минимизация стоимости комплекса работ при заданном времени выполнения проекта. *Комплексная оптимизация* представляет собой нахождение оптимального соотношения величин стоимости и сроков выполнения проекта в зависимости от конкретных целей, ставящихся при его реализации.

Вначале рассмотрим анализ и оптимизацию календарных сетей, в которых заданы только оценки продолжительности работ.

Анализ сетевого графика начинается с анализа топологии сети, включающего контроль построения сетевого графика, установление целесообразности выбора работ, степени их расчленения.

Затем проводятся классификация и группировка работ по величинам резервов. Следует отметить, что величина полного резерва времени далеко не всегда может достаточно точно характеризовать, насколько напряжённым является выполнение той или иной работы некритического пути. Всё зависит от того, на какую последовательность работ распространяется вычисленный резерв, какова продолжительность этой последовательности.

Определить степень трудности выполнения в срок каждой группы работ некритического пути можно с помощью коэффициента напряжённости работ.

Коэффициентом напряжённости работы называется отношение продолжительности несовпадающих, но заключённых между одними и теми же событиями, отрезков пути, одним из которых является путь максимальной продолжительности, проходящий через данную работу, а другим — критический путь.

Этот коэффициент может изменяться в пределах от 0 (для работ, у которых отрезки максимального из путей, не совпадающие с критическим

путём, состоят из фиктивных работ нулевой продолжительности) до 1 (для работ критического пути).
Обратим внимание на то, что больший полный резерв одной работы (по сравнению с другой) не обязательно свидетельствует о меньшей степени напряжённости её выполнения. Это объясняется разным удельным весом полных резервов работ в продолжительности отрезков максимальных путей, не совпадающих с критическим путём.
Вычисленные коэффициенты напряжённости позволяют дополнительно классифицировать работы по зонам:

- критическая $К > 0,8$,
- подкритическая $0,6 < К < 0,8$,
- резервная $К < 0,6$.

Оптимизация сетевого графика представляет процесс улучшения организации выполнения комплекса работ с учётом срока его выполнения. Оптимизация проводится с целью сокращения длины критического пути, выравнивания коэффициентов напряжённости работ, рационального использования ресурсов.
В первую очередь принимаются меры по сокращению продолжительности работ, находящихся на критическом пути. Это достигается:

- перераспределением всех видов ресурсов, как временных (использование резервов времени некритических путей), так и трудовых, материальных, энергетических, при этом перераспределение ресурсов должно идти, как правило, из зон, менее напряжённых, в зоны, объединяющие наиболее напряжённые работы. Например, можно увеличить сменность работ на «узких» участках строительства. Это мероприятие наиболее эффективно, поскольку позволяет добиться нужного результата при тех же ведущих машинах (экскаваторе, станке и т.д.), только увеличив численность рабочих.

- сокращением трудоёмкости критических работ за счёт передачи части работ на другие пути, имеющие резервы времени;
- пересмотром топологии сети, изменением состава работ и структуры сети.
- обеспечить проведение параллельных (совмещенных) работ;
- разделить широкий фронт работ на более мелкие захватки или участки;
- уменьшить продолжительность программы можно путем изменения применяемой технологии, например, в строительстве, заменой монолитных железобетонных конструкций сборными, другими сборными элементами, изготавливаемыми на заводе.

Проводя корректировку графика надо иметь в виду, что рабочих насыщают ресурсами до определенного предела (чтобы каждый рабочий был обеспечен достаточным фронтом работ и имел возможность соблюдать правила техники безопасности).

В процессе сокращения продолжительности работ критический путь может измениться, и в дальнейшем процесс оптимизации будет направлен на сокращение продолжительности работ нового критического пути и так будет продолжиться до получения удовлетворительного результата. В идеале длина любого из полных путей может стать равной длине критического пути или по крайней мере пути критической зоны. Тогда все работы будут вестись с равным напряжением, а срок завершения проекта существенно сократится.

Самый очевидный вариант частной оптимизации сетевого графика с учётом стоимости предполагает использование резервов времени работ. Продолжительность каждой работы, имеющей резерв времени, увеличивают до тех пор, пока не будет исчерпан этот резерв или пока не будет достигнуто верхнее значение продолжительности. Продолжительность каждой работы целесообразно увеличить на величину такого резерва, чтобы не изменить ранние сроки наступления всех событий сети, то есть на величину свободного резерва времени.

На практике при попытках эффективного улучшения составленного плана неизбежно введение дополнительно к оценкам сроков фактора стоимости работ. Проект может потребовать ускорения его выполнения, что, естественно, отразится на стоимости: она увеличится. Поэтому необходимо определить оптимальное соотношение между стоимостью проекта и продолжительностью его выполнения.

При использовании метода «время–стоимость» предполагают, что уменьшение продолжительности работы пропорционально возрастанию её стоимости. Возрастание стоимости при уменьшении времени называется *затратами на ускорение*.

Весьма эффективным является использование метода статистического моделирования, основанного на многократных последовательных изменениях продолжительности работ (в заданных пределах) и «проигрывании» на компьютере различных вариантов сетевого графика с расчётами всех его временных параметров и коэффициентов напряжённости работ.

Например, можно взять в качестве первоначального план, имеющий минимальные значения продолжительности работ и, соответственно, максимальную стоимость проекта. А затем последовательно увеличивать продолжительность выполнения комплекса работ путём увеличения продолжительности работ, расположенных на некритических, а затем и на критическом (критических) пути до удовлетворительного значения стоимости проекта. Соответственно, можно взять за исходный план, имеющий максимальную продолжительность работ, а затем последовательно уменьшать их продолжительность до такого приемлемого значения продолжительности проекта.

Процесс «проигрывания» продолжается до тех пор, пока не будет получен приемлемый вариант плана или пока не будет установлено, что все имеющиеся возможности улучшения плана исчерпаны и поставленные перед разработчиком проекта условия невыполнимы.

В настоящее время на практике сеть вначале корректируют по времени, т. е. приводят ее к заданному сроку окончания строительства. Затем приступают к корректировке графика по критерию распределения ресурсов, начиная с трудовых ресурсов.

Следует заметить, что при линейной зависимости стоимости работ от их продолжительности задача построения оптимального сетевого графика может быть сформулирована как задача *линейного программирования*, в которой необходимо минимизировать стоимость выполнения проекта при ограничении, во-первых, продолжительности каждой работы в установленных пределах, а, во-вторых, продолжительности любого полного пути сетевого графика не более установленного срока выполнения проекта.

6.1.7 Построение сетевого графика в масштабе времени

В практике получили распространение сетевые графики, составленные в масштабе времени с привязкой к календарным срокам. При контроле над ходом работ такой график позволит быстро найти работы, выполняемые в определённый период времени, установить их опережение или отставание и в случае необходимости перераспределять ресурсы.

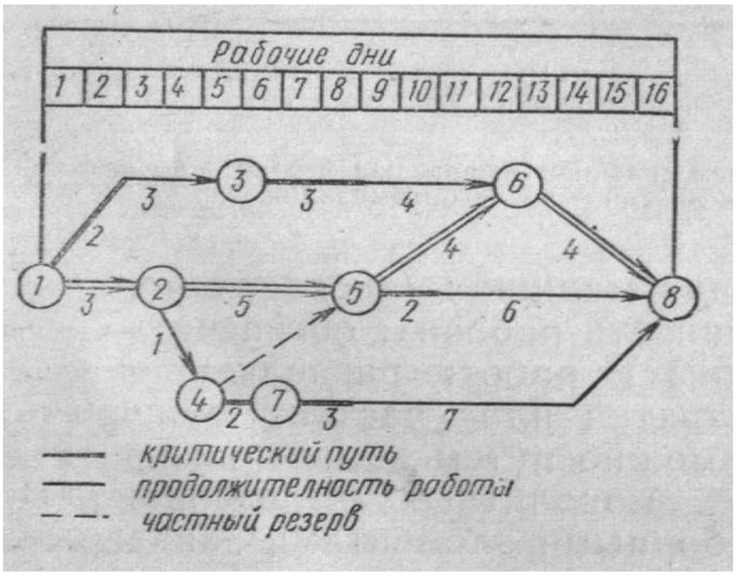


Рисунок 6.10- Сетевой график в масштабе времени

В случаях изменений исходных данных и фактического хода работ, сетевой график, составленный применительно к масштабу, вызывает усложнения при его корректировке. Поэтому такой метод применим для сравнительно небольших сетевых графиков.

6.2. Методы решения задач сетевого планирования

Существуют разные методы сетевого планирования. Модели, в которых взаимная последовательность и продолжительности работ заданы однозначно, называются детерминированными сетевыми моделями. К наиболее популярным детерминированным моделям относятся метод построения диаграмм Ганта и метод критического пути (CPM).

Если о продолжительности каких-то работ заранее нельзя задать однозначно или если могут возникнуть ситуации, при которых изменяется запланированная заранее последовательность выполнения задач проекта, например, существует зависимость от погодных условий, ненадежных поставщиков или результатов научных экспериментов, детерминированные модели неприменимы. Чаще всего такие ситуации возникают при планировании строительных, сельскохозяйственных или научно-исследовательских работ. В этом случае используются вероятностные модели, которые делятся на два типа:

- неальтернативные – если зафиксирована последовательность выполнения работ, а продолжительность всех или некоторых работ характеризуется функциями распределения вероятности;
- альтернативные – продолжительности всех или некоторых работ и связи между работами носят вероятностный характер.

К наиболее распространенным методам вероятностного сетевого планирования относятся:

- метод критического пути (CPM);
- метод оценки и пересмотра планов (PERT);
- метод имитационного моделирования или метод Монте-Карло;
- метод графической оценки и анализа (GERT).

6.2.1 Диаграмма Ганта и циклограмма

Одним из наиболее распространенных способов наглядного представления производственного процесса или проекта во времени является линейный или ленточный календарный график – Диаграмма Ганта. Диаграмма Ганта – горизонтальная линейная диаграмма, на которой задачи проекта представляются протяженными во времени отрезками, характеризующимися датами начала и окончания, задержками и, возможно, другими временными параметрами. Диаграмма Ганта представляет собой график, в котором процесс представлен в двух видах. В левой части проект представлен в виде списка задач (работ, операции) проекта в табличном виде с указанием названия задачи и длительности ее выполнения, а часто и работ, предшествующих той или иной задаче. В правой части каждая задача проекта, а точнее длительность ее выполнения, отображается графически, обычно в виде отрезка определенной длины с учетом логики выполнения задач проекта (рисунок 6. 11).

В верхней, правой части диаграммы Ганта располагается шкала времени. Длина отрезка и его расположение на шкале времени определяют время начала и окончания каждой задачи. Кроме того, взаимное расположение отрезков задач показывает, следуют ли задачи одна за другой или происходит их параллельное выполнение.

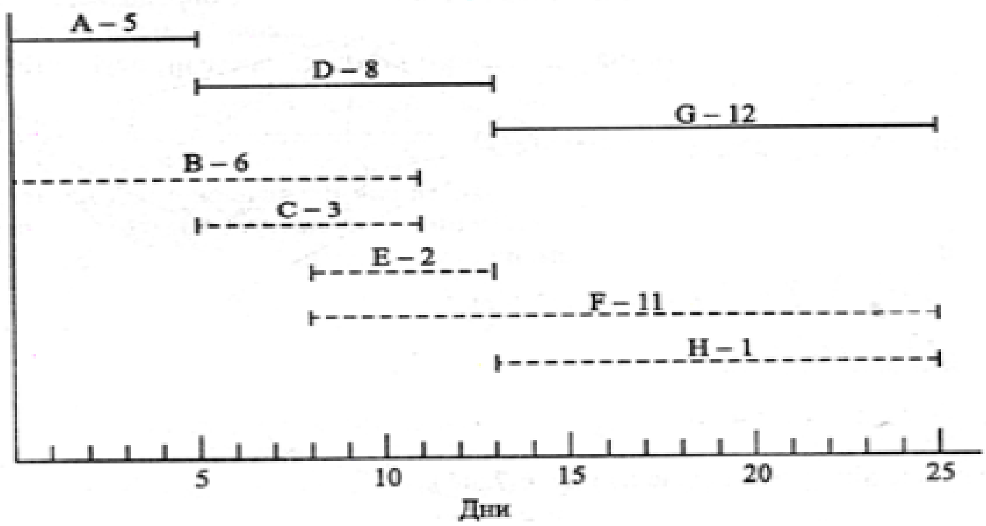


Рисунок 6.11. Диаграмма Ганта

Циклограмма представляет собой линейную диаграмму продолжительности работ, которая отображает работы в виде наклонной линии в двухмерной системе координат, одна ось которой изображает время, а другая – объемы или структуру выполняемых работ. Существуют циклограммы ритмичного и неритмичного потока. (рис 4.5, рис4.6). Равноритмичным потоком называют такой поток, в котором все составляющие потоки имеют единый ритм, т.е. одинаковую продолжительность выполнения работ на всех захватках

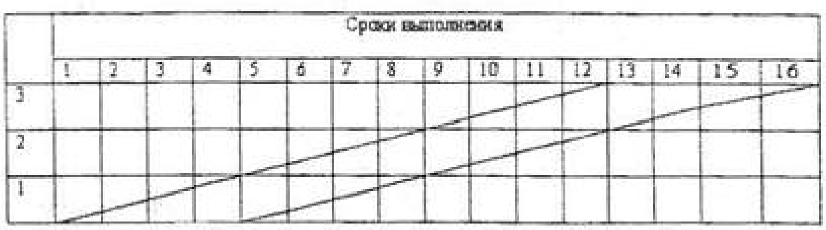


Рисунок 6.12. Равноритмичная циклограмма

Рисунок 6.13. Циклограмма неритмичного потока

В настоящее время циклограммы практически не используются в управленческой практике как по причине недостатков, указанным ниже, так и по причине неактуальности поточного строительства. Эти модели просты в исполнении и наглядно показывают ход работы. Однако они не могут отразить сложности моделируемого процесса – форма модели вступает в противоречие с ее содержанием. Основными недостатками являются:

- отсутствие наглядно обозначенных взаимосвязей между отдельными работами (зависимость работ, положенная в основу графика, выявляется только один раз в процессе составления графика (модели) и фиксируется как неизменная; в результате такого подхода заложенные в графике технологические и организационные решения принимаются обычно как постоянные и теряют свое практическое значение после начала их реализации);
- негибкость, жесткость структуры линейного графика, сложность его корректировки при изменении условий (необходимость многократного пересоставления графика, которое, как правило, из-за отсутствия времени не может быть выполнено);
- невозможность четкого разграничения ответственности руководителей различных уровней (информация, поступившая о ходе разработки, содержит в себе на любом уровне слишком много сведений, которые трудно оперативно обработать);
- сложность вариантной проработки и ограниченная возможность прогнозирования хода работ [7].

6.2.2 Метод критического пути (CPM)

Метод критического пути позволяет рассчитать возможные календарные графики выполнения комплекса работ на основе описанной логической структуры сети и оценок продолжительности выполнения каждой работы, определить критический путь для проекта в целом. В основе метода лежит определение наиболее длительной последовательности задач от начала проекта до его окончания с учетом их взаимосвязи. Метод критического пути исходит из того, что длительность операций можно оценить с достаточно высокой степенью точности и определенности. Основным достоинством метода критического пути является возможность манипулирования сроками выполнения задач, не лежащих на критическом пути. Конечным результатом применения метода критического пути (CPM) будет построение временного графика выполнения проекта. Для этого проводятся специальные вычисления. Каждому процессу $w \in W$ сопоставим положительное время его выполнения $t_W > 0$, фиктивным процессам сопоставим время выполнения равное 0 . Процесс не может до окончания предшествовавших процессов. Значит продолжительность выполнения проекта не может превышать сумму продолжительностей работ по некторому пути, соединяющему начальную и конечную вершины. *Сумма продолжительностей процессов входящитх в путь называется длиной пути. Критическим путем назовем путь максимальной длины.*

Для вычисление критической пути для сетевого графика каждому событию (вершине графа) сопоставляется число Θ_i называемая *ранним наступлением события* i. Это минимальное значение времени в которое событие i может прозойти (процесс с началом в этой вершине должен начаться не раньше, момента Θ_i). Ранние начала событияй определяются из соотношениия

$$\Theta_i = \max\{t_j + \Theta_{beg\,j} \mid end\, j = i\}$$

Иначе говоря, если для узла i узлы $p, q, ..., v$ связаны с узлом i процессами

$$w_1 = (p,i), w_2 = (q,i), ..., w_k = (v,i)$$

то $\Theta_i = \max\left\{\Theta_p + t_{w_1}, \Theta_q + t_{w_2}, ..., \Theta_v + t_{w_k}\right\}$.

Причем $\Theta_{i0} = 0$. Рассмотрими алгоритми вычислания ранних наступлений событий. множество вершин M разобьем на 3 подмножества $M = M_0 \cup M_1 \cup M_2$, где M_0 – множество вершин с вычисленными, M_1 – с вычисляемыми ранними временами наступления событий, а множество M_2 – вершины еще не затронутые вычислительным процессом. Каждой вершине i сопоставим r_i – число еще не просмотренных дуг с концом в i. Начальное состояние алгоритма следующее:

$M_0 = \varnothing \quad M_1 = \{i_0\} \quad M_2 = M \setminus M_1$

$\Theta_i = 0$, $r_i = |\{j \mid end\, j = i\}|$, $i \in M$

$\Theta_{i0} = 0$

Шаг алгоритма:

1. Если множество M_1 пусто, то завершить вычисления.
2. Если нет, то выбрать в M_1 произвольную вершину (обозначим ее i_1). Перевести эту вершину в $M_0(M_0 := M_0 \cup \{i_1\},\ M_1 = M_1 \setminus \{i_1\})$.
3. Для каждой дуги w выходящей из i_1 , и ее конца $i_2 = end\,w$ выполнить следующее $\Theta_{i_2} = \max\{\Theta_{i_2}, \Theta_{i_1} + t_w\}$. После чего r_{i_2} уменьшить на 1, и если $r_{i_2} = 0$ то перевести i_2 из M_2 в $M_1(M_0 := M_0 \cup \{i_1\},\ M_1 = M_1 \setminus \{i_1\})$.

Назовем поздним временем наступления события i число Δ_i –самое позднее возможное время наступления события i, то есть максимальное значение времени в которое события i может прозойти (процесс с концом в этой вершине не должен закончиться не позже момента Δ_i). Позднее время наступления события определябтся исходя из соотношения

$$\Delta_i = \min\{\Delta_{end\,j} - t_j \mid beg\,j = i\}$$

Иначе говоря, если для узла i узлы $p, q, ..., v$ связаны с узлом i процессами

$$w_1 = (p,i), w_2 = (q,i), ..., w_k = (v,i),$$

то $\Delta_i = \min\{\Delta_p - t_{w_1}, \Delta_q - t_{w_2}, ..., \Delta_v - t_{w_k}\}$. В обоих соотношениях пологаем $\Delta_{i_d} = \Theta_{i_d}$, пологая , что самое позднее время и самое раннее время завершения проекта совподают, где i_d – конечная $(destination)$ вершина. Рассмотрим следующий пример (Рис.6.14):

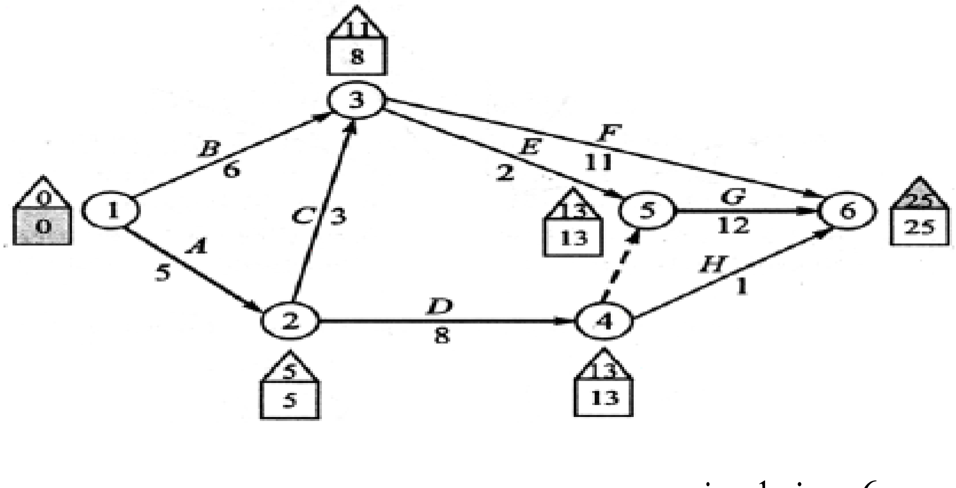


Рисунок 6.14. Определение критической пути. Здесь $i_0 = 1,\ i_d = 6$.

Для данного графа справедливы следующие соотношения:

1. Ранние наступления событий.

$$\Theta_2 = \Theta_1 + t_{12} = 5$$
$$\Theta_3 = \max\{\Theta_1 + t_{13}, \Theta_2 + t_{23}\} = \Theta_2 + t_{23} = 8$$
$$\Theta_4 = \Theta_2 + t_{24} = 13$$
$$\Theta_5 = \max\{\Theta_3 + t_{35}, \Theta_4 + t_{45}\} = \Theta_4 + t_{45} = 13$$
$$\Theta_6 = \max\{\Theta_3 + t_{36}, \Theta_4 + t_{46}, \Theta_5 + t_{56}\} = \Theta_5 + t_{56} = 25$$

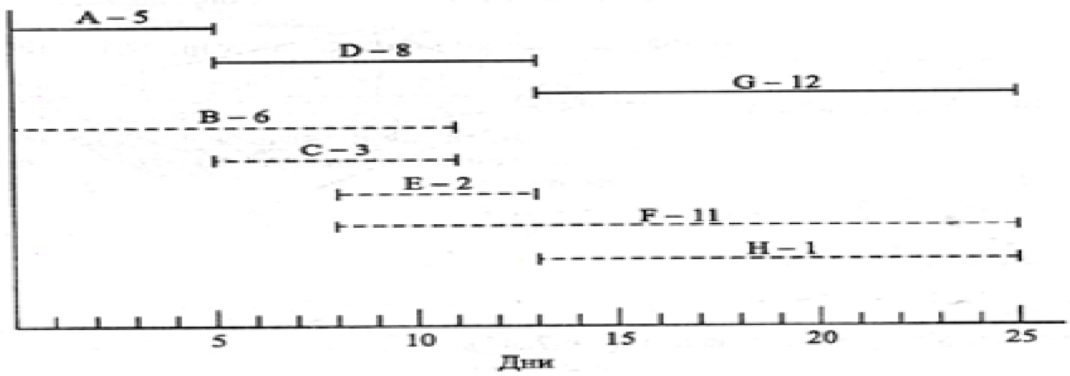


Рисунок 6.15. Временной график

2. Поздние наступление событий

$$\Delta_6 = \Theta_6 = 25$$
$$\Delta_5 = \Delta_6 - t_{56} = 13$$
$$\Delta_4 = \min\{\Delta_6 - t_{46}, \Delta_5 - t_{45}\} = \Delta_5 - t_{45} = 13$$
$$\Delta_3 = \min\{\Delta_6 - t_{36}, \Delta_5 - t_{35}\} = \Delta_5 - t_{35} = 11$$
$$\Delta_2 = \min\{\Delta_4 - t_{24}, \Delta_3 - t_{23}\} = \Delta_4 - t_{24} = 5$$
$$\Delta_1 = \min\{\Delta_3 - t_{13}, \Delta_2 - t_{12}\} = \Delta_2 - t_{12} = 0$$

Правильные вычисления должны были дать результат $\Delta_{i_0} = 0$. Критический путь составляет процессы $1 \to 2 \to 4 \to 5 \to 6$.

Для рассмотренного проекта составим временное график (рис.6.15). Критические процессы обозначены сплошными линиями, и стыкуются без перекрытий. Некритические процессы показаны пунктирными линиями, и с началом их выполнения необходимо определится.

6.3. Потоки в сети. Алгоритм Форда-Фалкерсона

Пусть дана сеть, имеющая один источник (пункт А) и ровно один сток (пункт Б). Предположим, что эта сеть представляет собой схему транспортных потоков. Дуги сети изображают допустимые напраления движения, вершины – пункты, в которой можно сменить направления движения, а числа, поставленные в соответствие дугам, интерпретируются как пропускные способности дуг, т.е. как количество едениц груза, которые можно пропустить по данной дуге за фиксированную еденицу времени. Предпологая, что из пункта А в пункт Б происходит непрерывный грузопоток, рассмотрим два естественных вопроса:

● какое минимальное число единиц груза можно доставлять в Б за единицу времени и как нужно для этого распределить грузы между направлениями движения?

●какие участки сети являютяс «узким местом»? Иначе говоря, пропускную способность какой дуги (дуг) надо увеличить, чтобы количество доставляемых грузов увеличилось?

В дальнейшем мы рассматриваем только те сети (G,α), в которых

- имеется ровно один источник s и один сток t;
- любая вершина достижима из источника и из любой вершины достижим сток

Число $\alpha(u,v)$будем называть пропусконой спсобностью дуги (u,v).

Определение 6.1. *Функция φ с неотрицательными значениями, заданная на дугах сети (G,α) называется* ***потоком,*** *если*

1) для всякой дуги (u,v) выполнено неравенство $\varphi(u,v)\le\alpha(u,v)$;

2) выполнено условие сбалансированности потока: для всякой вершины сети, отличной от источника и стока, сумма значений потока на входящих дугах равна сумме значений потока на исходящих дугах.

Отметим, что

● вместо «значения потока на дуге» обычно говорят «поток вдоль дуги»;

● в любой сети можно задать нулевой поток, равный нулю вдоль всех дуг.

Пример потока. Изображая поток в сети, мы будем указывать рядом с каждой дугой два числа формата $x(y)$. Число x обозначает пропускную способность дуги, а y – поток вдоль дуги. Пример потока в сети приведен на рисунке 6.16.

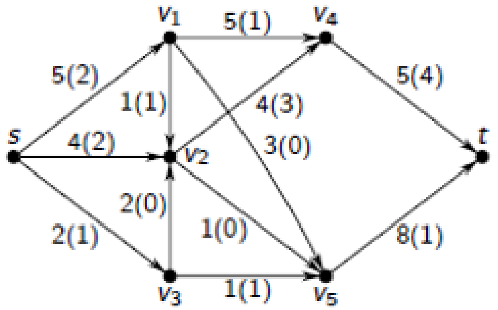


Рисунок 6.16. Потоки в сети.

Величина потока. Максимальный поток.

Определение 6.2. *Сумма потоков вдоль дуг, исходящего из источника, называется величиной потока и обозначается* $|\varphi|$ *для потока* φ. *Поток назывется максимальным, если его величина максимально среди всех потоков в данной сети*

Лемма 6.1. *В любой сети сумма потока вдоль дуг, исходящих из источника, равна сумме потоков вдоль дуг, входщих в сток.*

◄ Пусть G – сеть с множеством вершин $\{v_1, v_2, ..., v_v\}$, источником $v_1 = s$ и стоком $v_n = t$. Для любого $i = \overline{1,n}$ обозначим через $\varphi^+(v_i)$ сумму потоков вдоль всех дуг, входящих в v_i, а через $\varphi^-(v_i)$ – сумму потоков вдоль всех дуг, выходящих из v_i. Обозначим через S сумму потоков вдоль всех дуг сети. Ясно, что

$$\sum_{i=1}^{n} \varphi^-(v_i) = S = \sum_{i=1}^{n} \varphi^+(v_i)$$

.

По определению потока $\varphi^-(v_i) = \varphi^+(v_i)$ для всех . Так как

$\varphi^-(v_n) = \varphi^+(v_1) = 0$, получаем $\varphi^-(v_1) = \varphi^+(v_{n.}) = 0$ ►

Ввиду доказанной леммы, первый вопрос поставленный в начале на языке орграфов звучит как «найти величину максимального потока и сам максимальный поток».

Определение 6.3. Разрезом в сети (G, α) *с источником* s *и стоком* t *называется множество* $\{l_1, l_2, ..., l_k\}$ *дуг сети, обладающие следующими свойствами: 1) в орграфе* $G - \{l_1, l_2, ..., l_k\}$ *нет* (s,t)*-цепей; 2) если какую либо дуг из* $l_1, l_2, ..., l_k$ *добавить к орграфу* $G - \{l_1, l_2, ..., l_k\}$, *то в полученном орграфе найдется* (s,t)*-цепь.*

Например, множество всех дуг, исходящих из источника является разрезом в сети.

Определение 6.4. Пропускной способностью *разреза называется сумма пропускных способностей дуг, входящих в разрез. Разрез* *называется минимальным, если его пропускная способность минимальна среди пропускных способностей всех разрезов сети.*

Следующий основной результат о связах потоков и разрезов мы приводим без доказательства.

Теорема Форда-Фалкерсона. *В любой сети величина максимального потока равна пропускной способности минимального разреза.*

Эта теорема показывает, что содержательной точки зрения минимальный разрез – это и есть упомянутое в начале «узкое место» сети: увеличив пропускную способность любой дуги, входящий в минмимальный разрез, мы увеличим максимальный поток в сети.

Теперь мы можем сформулировать задачу о максимальном потоке на языке орграфов: *в заданной сети найти максимальный поток и минимальный разрез.*

Для построения максимального потока в сети нужно вначала понять, как увеличить уже имеющийся поток Основная хитрость в том, что: чтобы увеличить поток в сети,может потребоваться поток вдоль некоторых дуг уменьшить.

Рассмотрим сеть на рисунке 6.17. Поток, указанный на рисунке слева, не является максимальным. Если мы хотим его увеличить, пустив поток вдоль дуги (s,y), то вершина y «переполнится», ведь пропускная способность единственной исходящей из дуги y равна 1. Справится с этим переполнением можно только одним способом – уменьшить поток вдоль дуги (x,y), перенаправив его вдоль дуги (x,t). В результате этого действия мы и получим поток, изображенный на рисунке 6.17 справа – очевидна, максимальна для данной сети.

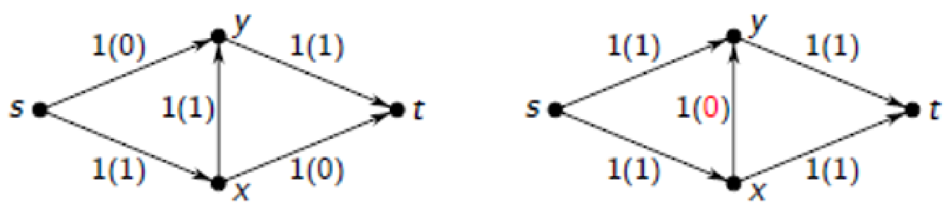


Рисунок 6.17. Увеличение потока.

Чтобы находить возможность увеличение потока, в таких случаях, вводится вспомогательная конструкция – остаточная сеть.

Определение 6.5. *Остаточной сетью* $(\hat{G},\alpha)$ *называется сеть полученная из сети* (G,α) *с заданным потоком* φ *применением двух* *преобразований:*

1)для каждой пары вершин $(u,v)\in V(G)$ *таких,что* $(u,v)\in E(G)$ *и* $(v,u)\notin E(G)$, *добавить к орграфу* G *дугу* (u,v) *пропускной способности 0;*

2) для каждой дуги $(u,v)\in E(G)$ *такой, что* $\varphi(u,v)>0$, *уменьшить пропускную способность дуги* (u,v) *на* $\varphi(u,v)$ *и увеличить пропускную способность дуги* (v,u) *на* $\varphi(u,v)$.

Остаточная сеть для предыдушего примера выглядит как на рисунке 6.18. (криволинейные дуги – добавленные дуги). Заметим, в ней есть цепь $s\xrightarrow{1}y\xrightarrow{1}x\xrightarrow{1}t$.

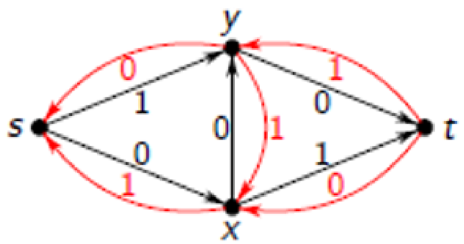


Рисунок 6.18. Остаточная сеть.

Определение 6.6. *Пропускной способностью цепи в сети называется минимум из пропускных способностей входящих в нее дуг.*

Лемма 6.2. *Если в остаточной сети* $(\hat{G},\alpha_\varphi)$ *существует простая* $(s,t)-$*цепь пропускной способности* $\delta>0$ *, то величину потока в сети* (G,α) *можно увеличить на* δ.

◄ Пусть C – упомянутая $(s,t)-$цепь. Изменим значения потока на дугах сети (G,α) по следующим правилам:

1) $\varphi'(u,v)=\varphi(u,v)+\delta$, *если* $(u,v)\in C$;

2) $\varphi'(u,v)=\varphi(u,v)-\delta$, *если* $(v,u)\in C$

3) $\varphi'(u,v)=\varphi(u,v)$ *остальных случаях*

Докажем, что φ' – поток и $|\varphi'|=|\varphi|+\delta$. Так как $\delta\le\alpha_\varphi(u,v)$ для любой дуги $(u,v)\in C$, то по определению остаточной сети для правила 1) имеем $\varphi(u,v)+\delta\le\alpha(u,v)$, а для правила 2), соответственно $\varphi(u,v)-\delta\ge 0$. С учетом того, что $\varphi-$ поток, получаем, что функция φ' неотрицательна и не превосходит α.

Проверим выполнение условия сбалансированости потока для φ'. Так как это условие выполнено для φ, равенство $\varphi^+(v)=\varphi^-(v)$ выполнено для любой вершины $v\in V(G)$, отличной от s и t. Если v не входит в цепь C, то поток вдоль дуг, инцендентных v, не изменится и равенство сохраняется. В противном случае в C есть пара дуг (u,v) и (v,w). При изменении потока вдоль дуги (u,v) по правилу 1) левая часть равенства увеличивается на δ, а при изменении по правилу 2) уменьшается на δ. Аналогично, при изменении потока вдоль (v,w) либо правая часть увеличивается на δ, либо левая часть уменьшается на δ; в любом случае, равенство будет восстановлено, т.е. условие балансированности выполнено. И так φ' – поток по поределению.

Поскольку C – простая цепь, в ней ровно одна исходящая из s дуга и нет дуг, входящих в s. Увеличение потока вдоль дуги, исходящей из s, происходит согласно правилу 1). Значить

$$|\varphi'| = \varphi'^{-}(s) = \varphi^{-}(s) + \delta = |\varphi| + \delta. \blacktriangleright$$

Применив доказанную лемму к нашему примеру в остаточной сети на рисунке 6.18. цепь $s \xrightarrow{1} y \xrightarrow{1} x \xrightarrow{1} t$, увеличим на единицу поток вдоль дуг (s,y) и (x,t) уменьшим на единицу поток вдоль (x,y), и в результате получим максимальный поток в исходной цепи (см. рис.6.16 справа)

Алгоритм Форда-Фалкерсона

Изложим алгоритм решения задачи о максимальном потоке

Вход: сеть (G,α) с источником s и стоком t.

Выход: максимальный поток φ *и минмиальный разрез* R *в сети* (G,α).

1. Для всякой дуги (u,v) положить $\varphi(u,v) = 0$.
2. Построить остаточную сеть $\hat{G}(\alpha_\varphi)$.
3. Если в $(\hat{G},\alpha_\varphi)$ найдется $(s,t)-$цепь C с пропускной способностью $\delta > 0$, изменить поток φ вдоль C по правилам 1)-3) и вернутся на шаг 2.
4. Найти множество S, состоящее из всех вершин v сети, для которых в $(\hat{G},\alpha_\varphi)$ существует $(s,v)-$цепь ненулевой пропускной способности; положить $R = \{(u,v) \in E(G) \mid u \in S, v \notin S\}$.

По лемме об остаточной сети, на каждой итерации шагов 2 и 3 алгоритм строит поток строо большей величины, чем предыдуший. Если все пропускные способности – целые числа, то число итераций не превосходит суммы пропускных способностей дуг, исходящих из источника. В частности алгоритм рано или поздно выполнит шаг 4 и остановится.

Докажем, что выход алгоритма Форда Фалкерсона являются именно максимальный поток и минимальный разрез.

Заметим, что $t \notin S$, поскольку иначе алгоритм перейдет на шаг 2 вместо шага 4; отсюда следует, что $R \neq \varnothing$. Пусть $T = V(G) \setminus S$.

Если $(u,v) \in E(G), u \in T$ и $v \in S$, то $\varphi(u,v) = 0$. В то же время

$\varphi(u,v)=\alpha(u,v)$ для каждой из дуг множества R, так как в остаточной сети эти дуги имают нелевую пропускную способность по определению множества R. Следовательно для любой дуги $(u,v)\in R$ из вершины v достижим сток t. Действительно, $\varphi(u,v)>0$, а значит, существует $(v,t)-$ цепь, вдоль всех дуг который поток положителен;по доказанному выше, в этой цепи нет вершин из S. В итоге R - разрез по определению, так как в орграфе нет $(s,t)-$цепей, а добавление к нему дуги (u,v) такую цепь обеспечивает.

Мы доказали, что из T в S «ничеге не течет». Тогда ввиду свойства сбалансированности суммарный поток на дугах из S в T должен равнятся $|\varphi|$. Но эта же сумма является, по определению, пропускной способностью разреза R; ввиду теоремы Форда-Фалкерсона, поток должен быть максимальным, а разрез – минимальным.

Важным шагом реализации алгоритма Форда-Фалкерсона является проверка условия на шаге 3, т.е. поиск остаточной сети $(s,t)-$цепи ненулевой пропускной способности.

Следующая проверка находит кратчайшую по числу дуг цепь с требуемыми свойствами. Для каждой вершины v вычисляются $\delta(v)-$ пропускная способность найденной $(s,v)-$цепи и $P(v)$ – предшествующая вершина в этой цепи.

Процедура поиска $(s,t)-$цепи ненулевой пропускной способности.

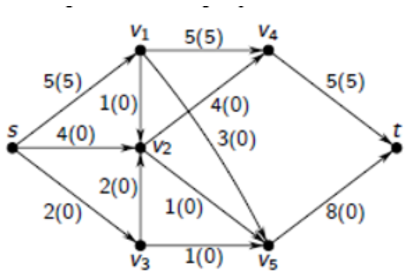
1. Положить $\delta(s)=\infty,\ \delta(v)=0$ для остальных вершин, а также $P(v)\neq\varnothing$ для всех вершин; создать очеред с одной вершины s.
2. Пока очередь не пуста, повторять
   -извлеч первый элемент из очереди (обозначим его через v);
   - для каждой дуги (v,w) такой, что $\alpha_\varphi(v,w)>0,\ \delta(w)=0,$ положить $\delta(w)=\min\{\delta(v),\alpha_\varphi(v,w)\}$; если $w=t$, закончить работу, иначе поместить w в конец очереди.

Если процедура завершила работу по условию $w=t$, то значения $P(t)$, $P(P(t))$,... определяют $(s,t)-$цепь с пропускной способностью $\delta(t)>0$.

Если же процедура закончилась ввиду исчерпания очереди, то такой цепи не существует, а вершина с условием $\delta(v)>0$ образует множество S, необходимое на шаге 4 алгоритма Форда-Фалкерсона.

Пример алгоритма Форда-Фалкерсона

1. Найдем максимальный поток и минимальный разрез в сети изображенной на рисунке 6.19.



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_2	s	4
v_3	s	2
v_4	v_2	4
v_5	v_2	1
v_1	v_4	4
t	v_5	1

Рисунок 6.19.

Рисунок 6.19.

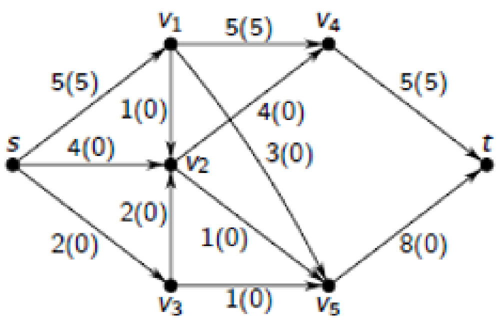
Эта же сеть с нулевым потоком изображена на рисунке 6.19 слева. Остаточную сеть будем представлять без рисовния в явном виде , запомнив,что

- по дугам можнодвигаться и против стрелок;
- пропускная способность дуги «против стрелки» равна второму числу в скобках, а по стрелке – разность первого и второго чисел.

Справа приведен «протокол» работы процедуры поиска $(s,t)-$цепи в остаточной сети (порядок строк соответствует порядку присвоения значения). В результате работы найдена цепь $s \to v_1 \to v_4 \to t$ пропускной спорсобностью 5 (см. выделенные значение в протоколе)

2. Увеличим поток вдоль дуг $(s,v_1),(v_1,v_4),(v_4,t)$ на $\delta(t)=5$ в

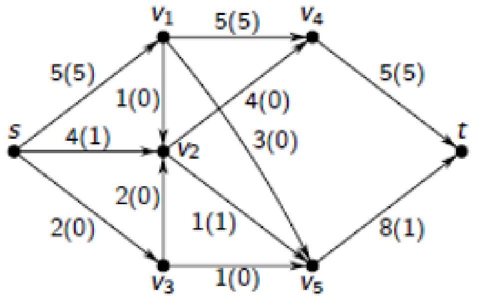
соответствии с правилом 1) увеличение потока. Получим поток изображенный на рисунке 6.20 слева. Протокол работы процедуры поиска $(s,t)-$цепи в получившейся остаточной сети приведен справа.Заметим, что вершина v_1 была достигнута из v_4 «против стрелки». В результате найдена цепь $s \to v_2 \to v_5 \to t$ пропускной способности 1.



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_2	s	4
v_3	s	2
v_4	v_2	4
v_5	v_2	1
v_1	v_4	4
t	v_5	1

Рисунок 6.20.

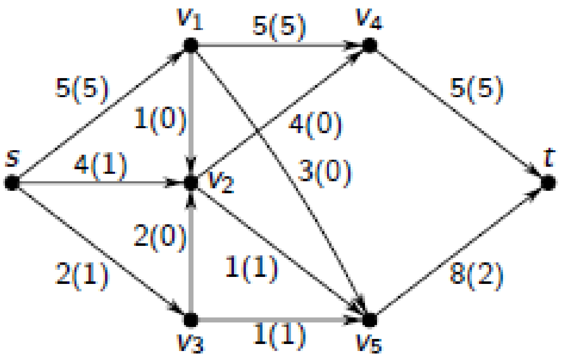
3. Увеличим поток вдоль дуг $(s,v_2),(v_2,v_5),(v_5,t)$ на $\delta(t)=1$ в соответствии с правилом 1) увеличение потока. Получим поток изображенный на рисунке 6.21 слева. Протокол работы процедуры поиска $(s,t)-$цепи в получившейся остаточной сети приведен справа. В результате найдена цепь $s \to v_3 \to v_5 \to t$ пропускной способности 1.



v	P(v)	δ(v)
s	∅	∞
v2	s	3
v3	s	2
v4	v2	3
v5	v3	1
v1	v4	3
t	v5	1

Рисунок 6.21.

4. Увеличим поток вдоль дуг $(s,v_3),(v_3,v_5),(v_5,t)$ на $\delta(t)=1$ в соответствии с правилом 1) увеличение потока. Получим поток изображенный на рисунке 6.22 слева. Протокол работы процедуры поиска $(s,t)-$цепи в получившейся остаточной сети приведен справа. В результате найдена цепь $s \to v_2 \to v_4 \to v_1 \to v_5 \to t$ пропускной способности 3.

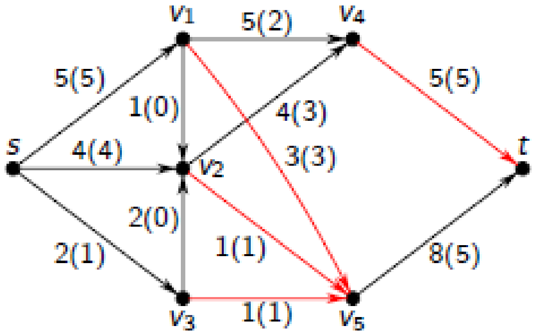


v	P(v)	δ(v)
s	∅	∞
v2	s	3
v3	s	1
v4	v2	3
v1	v4	3
v5	v1	3
t	v5	3

Рисунок 6.22.

Найденная $(s,t)-$цепь содержит обратную дугу (v_4,v_1); это значит, что поток вдоль дуги (v_4,v_1) можно уменьшить на 3, в соответствии правилам 2) увеличения потока, а поток вдоль дуг $(s,v_2),(v_2,v_4),(v_4,v_1),(v_1,v_5),(v_5,t)$ нужно увеличить на 3, в соответствии правилам 1) увеличения потока. Результат приведен на рисунке 6.23

5. Протокол работы процедуры поиска (s,t) – цепей в очередной



v	$P(v)$	$\delta(v)$
s	$\varnothing$	∞
v_3	s	1
v_2	v_3	1
v_4	v_2	1
v_1	v_4	1
очередь пуста		

Рисунок 6.23.

остаточной сети показывает, что таких цепей ненулевой пропускной способности нет и находит множество $S=\{s,v_1,v_2,v_3,v_4\}$. После этого на шаге 4 находим разрез $R=\{(v_1,v_5),(v_2,v_5),(v_3,v_5)(v_4,t)\}$ $S=\{s,v_1,v_2,v_3,v_4\}$. Пропускная способность R и величина максимального потока φ равны 10.

Контрольные вопросы по главе 6

1. Постановка задачи сетового планирования
2. Задачи сетового планирования
3. Основные понятия сетевого планирования
4. Порядок построения сетевых графиков
5. Правила построения сетевых графиков
6. Упорядочение сетевого графика
7. Временные параметры сетевых графиков
8. Анализ и оптимизация сетевого графика
9. Построение сетевого графика в масштабе времени
10. Методы решения задач сетевого планирования
11. . Диаграмма Ганта
12. Циклограмма
13 Метод критического пути (CPM)
14. Потоки в сетях
15. Максимальный поток
16. Минимальный разрез
17. Остаточная сеть
18. Алгоритм Форда-Фалкерсона

7. ТРАНСПОРТНАЯ ЗАДАЧА В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ

В настоящее время транспортная задача линейного программирования широко применяется как и в теоретических разработках, так и в практике планирования различных экономических процессов.

7.1. Постановка транспортной задачи и ее математическая модель

Некоторый однородный продукт, сосредоточенный у m поставщиков A_i в количестве a_i единиц $(i=\overline{1,m})$ соответственно, необходимо доставить n потребителям B_j в количестве b_j единиц $(j=\overline{1,n})$. Известна стоймость C_{ij} перевозки единицы груза от i-го поставшика к j-му потребителю.

Необходимо составить план перевозок, позволяющий вывезти все грузы, полностью удовлетворить потребность и имеющий минимальную стоймость.

Обозначим через x_{ij} количества единиц груза, запланированных перевозке от i-го поставшика к j-му потребителю; тогда условие задачи можно записать в виде таблицы (таблица 7.1.), в которую в дельнешем будем называть матрицей планирования.

Таблица 7.1.

Поставшики	Потребители				Запасы
	B_1	B_2	. . .	B_m	
A_1	C_{11} x_{11}	C_{12} x_{12}	. . .	C_{1n} x_{1n}	a_2
A_2	C_{21} x_{21}	C_{22} x_{22}	. . .	C_{2n} x_{2n}	a_2
. . .	. . .	. . .	. . .	. . .	. . .
A_m	C_{m1} x_{m1}	C_{m2} x_{m2}	. . .	C_{mn} x_{mn}	a_m
Потребности	b_1	b_2	. . .	b_n	$\sum a_i = \Sigma b_j$

Составим математическую модель задачи. Так как от i-го поставшика к j-му потребителю запланировано к перевозке x_{ij} единиц груза , то стоимость перевозки составит $C_{ij}x_{ij}$. Стоимость всего плана выразится двойной суммой:

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}x_{ij} \ .$$

Систему ограничений получаем из следующих условий задачи:
а) все грузы должны быть вывезены т.е.

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = \overline{1,m});$$

б) все потребносты должны быть удовлетворены т.е.

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = \overline{1,n}).$$

Таким образом математическая модель транспортной задачи имеет следующий вид: найти наименьшее значение линейной функции

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}x_{ij} \tag{7.1}$$

при ограничениях

$$\begin{cases} \sum_{j=1}^{n} x_{ij} = a_i, \ \ i = \overline{1,m} \\ \sum_{i=1}^{m} x_{ij} = b_j, \ j = \overline{1,n} \end{cases} \tag{7.2}$$

$$x_{ij} \geq 0, \ (i = \overline{1,m}), \ (j = \overline{1,n}) \tag{7.3}$$

В рассматриваемой модели, предполагается, что суммарные запасы равна суммарным потребностям т.е.

$$\sum a_i = \sum b_j \ . \tag{7.4}$$

Такая модель называется *закрытой*.

Теорема 7.1. *Любая транспортная задача, у которой суммарный обьем запасов совпадает с суммарным обьемом потребностей, имеет решение.*

◄ Пусть $\sum a_i = \sum b_j = M > 0$. Тогда величины $x_{ij} = a_i b_j / M$, $(i = \overline{1,m})$, $(j = \overline{1,n})$ являются планом, так как они удовлетворяют системе ограничений (7.2), (7.3). Действительно, подставляя значение x_{ij} в (7.2) и в (7.3) получим:

$$\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} \frac{a_i b_j}{M} = \frac{a_i}{M}\sum_{j=1}^{n} b_j = \frac{a_i}{M} M = a_i ,$$

$$\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} \frac{a_i b_j}{M} = \frac{b_j}{M} \sum_{i=1}^{m} a_i = \frac{b_j}{M} M = b_j .$$

Выберем из значений C_{ij} наибольшее $C' = \max C_{ij}$ и заменим в линейной функции все коэффициенты на C'; тогда учитывая (7.2) имеем:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} x_{ij} \le C^{/} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = C^{/} \sum_{i=1}^{m} a_i = C^{/} M .$$

Выберем из значений C_{ij} наименьшее $C^{//} = \min C_{ij}$ и заменим в линейной функции все коэффициенты на $C^{//}$; тогда учитывая (7.2) имеем:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} x_{ij} \ge C^{//} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = C^{//} \sum_{i=1}^{m} a_i = C^{//} M$$

Обьединяя два последных неравенства в одно, двойное получаем:

$$C^{//} M \le Z \le C^{/} M ,$$

т.е. линейная функция ограничена на множестве планов транспортной задачи. ►

7.2. Построение первоначального опорного плана

Как и для других задач линейного программирования, итерационный процесс по отысканию оптимального плана транспортной задачи начинают с нахождения опорного плана.

Рассмотрим систему ограничений (7.2) и (7.3) транспортной задачи. Она содержит mn неизвестных и $m+n$ уравнений, связанных соотношением (7.4). Если сложить посленно уравнения отдельно подсистемы (7.2) и отдельно подсистемы (7.3), то получим два одинаковых уравнения. В таблице 7.1 такое сложение равнозначно соответственно почленному сложению столбцов и почленному сложению строк. Наличие в системе ограничений двух одинаковых уравнений говорит о ее линейной зависимости. Если, одно из этих уравнений отбросить, то в общем случае система огограничений должгна содержать $m+n-1$ линейно независимых уравнений, следовательно, невырожденный опорный план транспортной задачи содержит $m+n-1$ положительных ькомпонент или перевозок.

Таким образом, если каким-либо способом получен невырожденный опорный план транспортной задачи, то в матрице $\left(x_{ij}\right)$ $(i=\overline{1,m})$, $(j=\overline{1,n})$ значений его компонент (таблица 7.1.) положительными являются только

$m+n-1$, остальные равны нулю.

Если условия транспортной задачи и ее опорный план записаны в виде таблицы 7.1, то клетки, в которых находятся отличные от нуля перевозки, называются *занятыми*, остальные – *незанятыми.* Занятые клетки соответствуют базисным неизвестным, и для невырожденного опорного плана их количество равно $m+n-1$. Если ограничения транспортной задачи записаны в виде (7.2) и (7.3) , то как известно, базисным неизвестным, включенным в опорный план соответствует система линейно независимых векторов. Опорность плана при записи условий транспортной задачи в виде таблицы заключается в его ацикличности, т.е. в таблице нельзя построить замкнутый цикл, все вершины которого лажат в занятых клетках.

Циклом *называется набор клеток вида* $(i_1j_1)(i_1j_2)\times(j_2i_2)...(j_1i_m)$, *в котором две и только две соседние клетки расположены в одном столбце или одной строке таблицы, причем последняя находится в той же строке или столбце, что и первая.*

Построение циклов начинают с какой-либо занятой клетки и переходят по столбцу (по строке) к другой занятой клетке, в которой делают поворот под прямым углом и движутся по строке (столбцу) к следующей занятой клетке и т.п. пытаясь возвращатся к первоначальной клетке. Если такой возврат возможен, то получен цикл и план не является опорным. Клетки, у которых происходит поворот под прямым углом, определяют вершины цикла. В противном случае план является опорным.

Всякий план транспортной задачи, содержащий более $m+n-1$ занятых клеток, не является опорным, так как ему соответствует линейно зависимая система векторов. При таком плане в таблице всегда построит замкнутый цикл, с помощью которого уменьшают и число занятых клеток до $m+n-1$. Если к занятым клеткам, определяющим опорный невырожденный план присоединит какую-либо незанятую клетку, то план становится не опорным, появляется единственный цикл, все вершины которого, за исключением одной, лежат в занятых клетках.

Сушествуют несколько простых схем построения первоначального опорного плана транспортной задачи, которые рассмотрим на примерах.

7.3. Метод северо-западного угла

Пусть условия транспортной задачи заданы табл. 7.2. Не учитывая стоймости перевозки единицы груза, начинаем удовлетворение потребностей первого потребителя B_1 за счет поставшика A_1. Для этого сравниваем $a_1=100$ с $b_1=200$, $a_1<b_1$, меньший из обемов, т.е. $a_1=100$ единиц зписываем в левый нижний угол клетки A_1B_1. Запасы первого поставшика полностью израсходованы, поэтому остальные клетки первой строки прочеркиваем. Потребность B_1 остались неудовлетворенным на 200-100=100 единиц. Сравниваем этот остаток с запасами поставшика A_2: так как $100<250$, то 100 единиц записываем в клетку A_2B_1, чем полностью удовлетворяем потребность потребителя B_1 а оставшиеся клетки в первом столбце прочеркиваем.

У поставшика A_2 осталось 150 единиц груза. Удовлетворим потребность B_2 за счет оставшегося у поставшика A_2 груза. Для этого сравниваем этот остаток с потребностями потребителя B_2 $:150 < 200$, записываем 150 единиц в клетку A_2B_2 и, так как запасы A_2 полностью израсходованы, прочеркиваем остальные клетки второй строки. Потребность B_2 остались неудовлетворенным на 50 единиц. Удовлетворяем их за счет поставшика A_3 и переходим к удовлетворению

Таблица 7.2

Постав-щики	Потребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10 / 100	7 / —	4 / —	1 / —	4 / —	100
A_2	2 / 100	7 / 50	10 / —	6 / —	11 / —	250
A_3	8 / —	5 / 50	3 / 100	2 / 50	2 / —	200
A_4	11 / —	8 / —	12 / —	16 / 50	13 / 250	300
Потреб-ности	200	200	100	100	250	850

B_3 за счет остатка , имеющегося у поставшика A_3, и т.п. Процесс продолжаем до тех пор, пока не удовлетворим всех потребителей за счет запасов помтавшиков. На этом построение первоначального опорного плана заканчивается.

Таким образом, в таблице 7.2 в правых верхних углах клеток стоят числа, определяющие стоимость перевозки единицы перевозки грузов, а в левых нижних углах – числа, определяющие план перевозок, так как их сумма по строкам равна запасам соответствующего поставшика, а сумма по столбцам – потребности соответствующего потребителя.

Проверим, является ли план, построенный в таблице 7.2, опорным. Видим, что, начиная движения от занятой клетки A_1B_1, вернуться не только в нее, но и любую другую занятую клетку, двигаясь только по занятым клеткам невозможно. Следовательно, план является опорным. В то же время план невырожденный, так как содержит точно $m+n-1=4+5-1=8$ занятых клеток.

При составлении первоначального опорного плана методом северо-западного угла стоимость перевозки единицы груза не учитывалась, поэтому построенный план далек от оптимального, получение которого связано с большими обьемом вычислительных рабат. Поэтому рассмотренный метод используется при вычислении с помощью компьютера.

Найдем общую стоимость составленного плана как сумму произведений обьемов перевозок, стоящих в левом углу занятых клеток,0 на соответствующие стоимости в этих же клетках:

$$Z = 100 \cdot 10 + 100 \cdot 2 + 150 \cdot 7 + 50 \cdot 5 + 100 \cdot 3 + 50 \cdot 2 + 50 \cdot 16 + 250 \cdot 13 = 6950 \text{ (ед.стоимости)}$$

Если при составлении опорного плана учитывать стоимость перевозки единицы груза, то план будет значительно ближе к оптимальному.

7.4. Метод минимальной стоимости

Суть метода заключается в том, что из всей таблицы стоимостей выбирают наименьшую и в клетку, которая ей соответствует, помещают меньшее из чисел a_i или b_j. Затем из расмотрения исключают либо строку, соответствующей поставшику, запасы которого полностью израсходованы, либо столбец, соответствующей потребителю, потребности которого полностью удовлетворены, либо столбец и строку , если израсходованы запасы поставшика и удовлетворены потребности потребителя. Из оставшейся части таблицы снова выбирают наименьшую стоимость, и процесс распределения запасов продолжаются, пока все запасы не будут распределены, а потребности удовлетворены.

Составим с помощью этого метода опорный план уже рассмотенной задачи. Заришем ее условия в твблицу (таблица 7.3). Выбираем в таблице наименьшую стоимость (это стоймость, помещенная в клетке A_2B_4), так как $a_1 = b_4$, 100 единиц груза помещаем в этой клетке и исключаем из рассмотрения первую строку и четвертый столбец. В оставшейся таблице стоимости наименьшей является стоимость, расположенная в клетке A_2B_1 и в клетке A_3B_5. Заполняем любую из них, например A_2B_1. Имеем $200 < 250$, следовательно, записываем в нее 200 и исключаем из рассмотрения столбец B_1. В клетку A_3B_5 записываем 200 и исключаем из рассмотрения строку A_3. В оставшейся таблице стоимости снова выбираем наименьшую стоимость и продолжаем процес до тех пор, пока все запасы не будут распределены, а потребности удовлетворены. В результате получен план

$$X = (x_{14} = 100; x_{21} = 200; x_{22} = 50; x_{35} = 200; x_{42} = 150; x_{43} = 100; x_{45} = 50;)$$

остальные значения переменных равна нулю.

План не содержит циклов и состоит из семи положительных перевозок, следовательно, является вырожденным планом.

Таблица 7.3

Поставшики	Потребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10 –	7 –	4 –	1 100	4 –	100
A_2	2 200	7 50	10 –	6 –	11 –	250
A_3	8 –	5 –	3 –	2 –	2 200	200
A_4	11 –	8 150	12 100	16 –	13 50	300
Потребности	200	200	100	100	250	850

Определим его стоимость:

$$Z = 100 \cdot 10 + 100 \cdot 2 + 50 \cdot 7 + 200 \cdot 2 + 150 \cdot 8 + 100 \cdot 12 + 50 \cdot 13 = 4300 \text{ (ед)}$$

Стоимость плана перевозок значительно меньше, следовательно ближе к оптимальному.

7.5. Метод двойного предпочтения

Если таблица стоимостей велика, то перебор всех элементов затруднителен. В этом случае используют метод двойного предпочтения, суть которого заключется в следующем.

В каждом столбце отмечают знаком $\vee$ клетку с наименьшей стоймостью. Затем то же проделывают в каждой строке. В резултате некоторые клетки имеют отметку $\vee\vee$. В них находятся минимальная стоимость как по столбцу так и по строке. В эти клетки помещают максимально возможные обьемы перевозок, каждый раз исключая из рассмотрения соответствуюшие столбцы и строки. Затем распределяют перевозки по клеткам, отмеченным знаком $\vee$. В оставшейся части таблицы стоимостей перевозки распределяют по наименьшей стоимости.

Применим метод двойного предпочтения к задаче, условия которой записаны в таблице 7.4.

Сначала заполняем клетки A_2B_1, A_1B_4, A_3B_5, а затем клетку A_4B_2. В оставшейся части таблицы последовательно заполняем клетки по минимальной стоимости A_2B_3, A_4B_3, A_4B_5. Полученный план в таблице 7.4

является вырожденным опорным планом. Найдем его стоимость:

$Z = 100 \cdot 10 + 200 \cdot 2 + 50 \cdot 10 + 200 \cdot 2 + 200 \cdot 8 + 50 \cdot 12 + 50 \cdot 13 = 4250$ (ед).

Таблица 7.4.

Поставшики	Пот ребители					Запасы
	B_1	B_2	B_3	B_4	B_5	
A_1	10 –	7 –	4 –	∨ ∨ 1 100	4 –	100
A_2	∨ ∨ 2 200	7 –	10 50	6 –	11 –	250
A_3	8 –	∨ 5 –	∨ 3 –	∨ 2 –	∨ ∨ 2 200	200
A_4	11 –	∨ 8 200	12 50	16 –	13 50	300
Потребности	200	200	100	100	250	850

Таким образом, наименьшую стоимость имеет опорный план, полученный методом двойного предпочтения, следовательно он наибелее близок к оптимальному плану.

7.6. Метод потенциалов

С помощью рассмотренных методов построения первоначального опорного плана можно получить вырожденный или невырожденный опорный план. Построенный опорный план транспортной задачи как и задачи линейного программирования можно было довести до оптимального с помощью симплек метода. Однако из-за громоздкости симплекс-таблиц, содержащих mn неизвестны, и большего обема вычислительных работ для получения оптимального плана используют более простые методы, самый распостраненый из которых метод потенциалов (модифицированный распределительный метод).

Теорема 7.2. *Если план $X^* = (x_{ij}^*)$ $X^* = (x_{ij}^*)$ транспортной задачи является оптимальным, то ему соответствует система из $m+n$ чисел U_i^* и V_j^*, удовлетворяющих условиям*

$$U_i^* + V_j^* = C_{ij} \quad \text{для } x_{ij}^* > 0,$$

$$U_i^* + V_j^* \le C_{ij} \quad \text{для } x_{ij}^* = 0$$

$$(i = \overline{1,m};\ j = \overline{1,n}).$$

Числа U_i^* и V_j^* называются потенциалами сответственно поставшиков *и потребителей.*

◀ Танспортную задачу минимизации линейной функции

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}x_{ij}$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = 1,2,...,m),$$

$$\sum_{j=1}^{n} x_{ij} = b_j \quad (j = 1,2,...,n)$$

$$x_{ij} \ge 0 \quad (i = 1,2,...,m;\ j = 1,2,...,n)$$

можно рассматривать как двойственную задачу некоторой исходной задачи линейного программирования, условия которой получают по общей схеме, рассмотренной первой главе, если каждому ограничению вида $x_{i1} + x_{i2} + ... + x_{in} = a_i$ в исходной задаче соответствует переменная U_i $(i = \overline{1,m})$, а каждому ограничению вида $x_{1j} + x_{2j} + ... + x_{mj} = b_j$ – переменная V_i $(j = \overline{1,n})$, а именно максимизировать линейную функцию $f = \sum_{i=1}^{m} a_i U_i + \sum_{j=1}^{n} b_j V_j$ при ограничениях $U_i^* + V_j^* = C_{ij}$ $(i = 1,2,...,m;\ j = 1,2,...,n)$. План X^* – оптимальный план двойственной задачи, поэтому план $Y^* = (U_i^*, V_j^*)$ является планом исходной задачи и на основании теоремы двойственности

$$\max f = \min Z,$$

или

$$\sum_{i=1}^{m} a_i U_i^* + \sum_{j=1}^{n} b_j V_j^* = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} x_{ij}^*,\ x_{ij}^* \ge 0.$$

На основаниии теоремы 7.1. получаем, что ограничения исходной задачи, соответствующие положительным компонентам оптимального плана двойственной задачи, удовлетворяются как строгие равенства, а соответствующим компонентам, равным нулю – как неравенства, т.е.

$$U_i^* + V_j^* = C_{ij} \quad \text{для}\ \ x_{ij}^* > 0,\ U_i^* + V_j^* \le C_{ij} \quad \text{для}\ \ x_{ij}^* = 0. \ ▶$$

На основании доказанной теоремы для того, чтобы первоначальный опорный план был оптимальным, необходимо выполнение следующих условий:

а) для занятой клетки сумма потенциалов должна быть равна стоимости еденицы перевозки, стоящей в этой клетке:

$$U_i + V_j = C_{ij} \tag{7.5}$$

б) для каждой незанятой клетки сумма потенциалов должна быть меньше или равна стоимасти единицы перевозки, стоящей в этой клетке

$$U_i + V_j \le C_{ij} \ . \tag{7.6}$$

Если хотя бы одна незанятая клетка не удовлетворяет условию (7.6), то опорный план является неоптимальным и его можно улучщить, вводя в базис вектор, соответствующий клетке, для которой нарушается условие оптимальности (т.е. в клетку надо переместить некоторое количества единиц груза).

Таким образом, для проверки плана на оптимальность необходимо сначала построить систему потенциалов.

Рассмотрим алгоритм метода потенциалов и одновременно проиллюстрируем его применение на опорном плане, полученном в таблице 7.3. Для этого к таблице добавим строку и столбец , которых записывается значения потенциалов (в результате получим таблицу 7.5).

1. *Построение системы потенциалов.* Для построение системы

потенциалов используем условие

$$U_i + V_j = C_{ij} \ ,$$

где C_{ij} – стоимость перевозки единицы груза занятой клетки в i-й стороке и j-м столбце.

Систему потенциалов можно построить только для невырожденного плана $m+n-1$ занятых клеток, поэтому для него можно составить систему из $m+n-1$ линейно независимых уравнений с $m+n$ неизвестными. Уравнений на одно меньше, чем неизвестных, поэтому система является неопределенной и одному неизвестному (обычно U_1) придают нулевые значения. После этого остальные потенциалы определяются однозначно.

Пусть неизвестен потенциал U_i; тогда из равенства (7.5) следует

$$V_j = C_{ij} - U_i \, .$$

Если известен потенциал V_j, то из того же равенства имеем:

$$U_i = C_{ij} - V_j \, .$$

Таким образом, для определения низвестно потенциала от величины C_{ij} занятой клетки следует вычесть известный потенциал.

В таблице 7.5 опорный план вырожденный, так как нехватает одной занятой клетки. Поэтому выбираем строку, которая содержит наибольшиее количество зханятых клеток (строка A_4) и пологаем $U_4 = 0$.

В строке A_4 три занятых клетки ($A_4B_2, A_4B_4A_4B_5$), которые связывают потенциал U_4 с потенциалами V_2, V_3, V_5. Определим эти потенциалы:

$V_2 = C_{42} - U_4 = 8 - 0 = 8$, $V_3 = C_{43} - U_4 = 12 - 0 = 12$, $V_5 = C_{45} - U_4 = 13\text{-}0\text{=}13$.

С помощью потенциала U_4 еще какой-нибудь потенциал невозможно, поэтому отмечаем его знаком $\vee$. Теперь по очередно просматриваем столбцы B_2, B_3 и B_5, для которых потенциалы уже определены. В столбце B_2 имеются две занятые клетки (A_4B_2 *и* A_4B_5), которые связывает потенциал V_2 с потенциалами U_2 и U_4; потенциал U_4 уже определен.

Таблица 7.5.

Матрица планировани я		Потребители					Запасы
		B_1	B_2	B_3	B_4	B_5	
Поставшики	v_j / u_i	$\vee$ $v_1 = 3$	$\vee$ $v_2 = 8$	$\vee$ $v_3 = 12$	$\vee$ $v_4 = 13$	$\vee$ $v_5 = 13$	
A_1	$\vee$ $u_1 = -12$	10 –	7 –	4 –	1 100	4 –	100
A_2	$\vee$ $u_2 = -1$	2 200	7 50 –	10 – 1	6 – + 6	11 – 1	250
A_3	$\vee$ $u_3 = -11$	8 –	5 –	3 –	2 – –	2 + 200	200
A_4	$\vee$ $u_4 = 0$	11 –	8 150 +	12 100	16 –	13 50 –	300
Потребности		200	200	100	100	250	850

Переходим к клетке A_2B_2 и с помощью C_{22} определим неизвестный потенциал: $U_2 = C_{22} - V_2 = 7 - 8 = -1$. Отмечаем потенциал V_2 знаком $\vee$ и переходим к столбцу B_3. В этом столбце нет занятых клеток, которые бы связали V_3 с неизвестными потенциалами строк. Отмечаем потенциал V_3 знаком $\vee$ и переходим к столбцу B_5. В нем одна занятая клетка (A_3B_5), которая связывает V_5 с неизвестным потенциалом U_3. Определим его: $U_3 = C_{32} - V_5 = 2 - 13 = -11$. Отмечаем потенциал V_3 знаком $\vee$ и, так как неизвестные потенциалы столбцов использованы, переходим к известным потенциалам строк, которые еще не отмечены знаком $\vee$, затем просматриваем соответствующие им строки.

Потенциал U_2 занятой клеткой A_2B_1 связан с потенциалом V_1. Находим этот потенциал: $V_1 = 2 - (-1) = 3$ и отмечаем потенциал U_2 знаком $\vee$.

В строке A_3 нет занятых клеток, которые связалы бы потенциал с U_3 неизвестным потенциалом столбца; потенциал U_3 отмечаем знаком $\vee$.

Переходим к известным потенциалам столбцов, которые не отмечены знаком $\vee$ (это потенциал V_1). Но в столбце B_1 нет занятых клеток, которые связывали бы V_1 с неизвестным ипотенциалом строки, поэтому потенциал V_1 отмечаем знаком $\vee$. Построение потенциалов прервалось, потенциалы U_1 и V_4 остались неопределенными. Это произошло потому, что опорный план вырожденный (отсутствует одна занятая клетка). Для устранения вырожденности дополняют количество занятых клеток до $m+n-1$, вводя нулевые перевозок. Клетки, в которые введены нулеве перевозки, называются *фиктивно* занятыми.

Чтобы определить поенциалы U_1 и V_4 необходимо сделать фиктивно занятой одну из незанятых клеток либо строки A_1, либо столбца B_4, для которых один из потепнциалов определен. Задача решается на минимизацию линейной функции, пэтому целесообразно фиктивно занятой клетку, которая стоит наименьшая стоимость.

Простматривая стоимость, стоящие в незанятых клетках стороки A_1 и столбца B_4, выбираем наименьшую ($\min C_{ij} = 2$), которая соответствует клетке A_3B_4, в нее записываем нуль и считаем занятой. Теперь клетка A_3B_4 связывает потенциал V_4 с потенциалом U_3; $V_4 = C_{34} - U_3 =$ 2-(-11) =13. Затем находим $U_1 = C_{14} - V_4 =$ 1-13=-12.

Система потенциалов построена, знаки $\vee$, которыми отмечались потенциалы следует стереть. Проверяем правильность построения системы.

Для этого просматриваем занятые клетки строк и для каждой из них определяем сумму потенциалов. Если для всех занятых клеток выполняется равенство (7.5), то система построена правильно. В противном случае ее надо построить заново или изменить так, чтобы условия (7.5) выполнялось.

2 Проверка выполнения условия оптимальности для незанятых клеток.

Просматриваем и для каждой незханятой клетки проверяем выполнение условия (7.6), т.е. суммируем потенциалы, на пересечениях, которого стоит незанятая клетка, сумму сравниваем со стоимостью, стоящей в ней. Если для всех незанятых клеток $U_i + V_j \le C_{ij}$, то план оптимальный. Если для некоторых клеток $U_i + V_j > C_{ij}$, то план является неоптимальным. Тогда для каждой клетки, в которой не не выполняется условие оптимальности, находим величину разности $(U_i + V_j) - C_{ij} > 0$ и записываем в ее значекние в левый нижний угол этой же клетки.

В таблице 7.5 для незанятых клеток получаем: для строки A_1: -9<10, -4<7, 0<4, -8<4; для строки A_2:для клетки A_2B_3 имеем 11>10 или 11-10=1; условие оптимальности нарушено, разность, равную единице записываем эту клетку; для клетки A_2B_4 имеем 11>6, 12-6=6; условие также нарушено, разность, равную шести, записываем эту клетку; для клетки A_2B_5 имеем 12>11, 12-11=1, разность, равную единице записываем в клетку; для строки A_3: -8<8, -3<5, 1<3; для строки A_4: 3<11, 13<16.

Таким образом, имеются три клетки, в которых нарушено условие оптимальности; разности соответственно равны 1, 6 и 1.

3. *Выбор клетки, в которую необходимо послать перевозку.*

Транспортная задача линейного программирования решается на минимумум ли нейной функции, поэтому алгоритм ее решения также следует общим принципам алгоритма симплексного метода решения задач на минимум.

Если отождествлять занятые клетки с векторами, составляющими базис, а незанятые – с остальными векторами системы ограничений, то в общей задаче линейного программирование в базис включается тот вектор, которому соответствует $\max\theta_0(Z_j - C_j)$. В транспортной задаче значение суммы Z_j заменено суммой потенциалов $(U_i + V_j)$, поэтому загрузке подлежит в первую очеред клетка, которойц соответствует $(U_i + V_j) - C_{ij}$.

Таким образом, в рассматриваемой примере $\max\{1,6,1\} = 6$, клетку A_2B_4 необходимо сделать занятой. Для этого сначала необходимо определить, сколько единиц груза должно быть перераспределено в нее.

4.Построения цикла и определение величины перераспределения груза. Для определение количества единиц груза, подлежащих перераспределению, отмечаем знаком «+» незанятую клетку, которую надо загрузить. Это означает, что клетка присоединяется к занятым клеткам. В таблице занятых клеток стало $m+n$, поэтому появляется цикл, все вершины которого, за исключением клетки отмеченной «+», находятся в занятых клетках, причем этот цикл единственный. Отыскиваем цикл и, начиная движение от клетки, отмеченной знаком «+» поочередно проставляем знаки «–» и «+». Затем находим $\theta_0 = \min x_{ij}$, где x_{ij} – перевозки, стоящие в вершинах цикла, отмеченных знаком «–». Величина θ_0 определяет сколько единиц груза можно перераспределить по найденному циклу. Значение θ_0 записываем в незанятую клетку, отмеченную знаком «+» двигаясь по циклу, вычитаем θ_0 из обьемов перевозок, расположенных в клетках, которые отмечены знаком «–», и прибавляем к обьемам перевозок, расположенных в клетках, которые отмечены знаком «+». Если θ_0 соответствютнесолько минимальных перевозок, то при вычитании оставляем в соответствующих клетках нулевые перевозки в таком количестве, чтобы во вновь полученном опорном плане занятых клеток было не более $m+n-1$.

В рассматриваемом примере клетку A_2B_4 отмечаем знаком «+» и находим цикл, приведенный в таблице 7.5. Имеем $\theta_0 = \min\{50;50;0\} = 0$, т.е. нулевую нагрузку необходимо переместить в клетку A_2B_4, остальные числа при вычмтании и прибавлении нуля, очевидно не измениться.

5. В результате перераспределения θ_0 получен новый опорный невырожденный план (таблица 7.6), который снова подлежит проверке на оптимальность. Для проверки на оптимальность нового опорного плана можно вновь построить систему потенциалов и проверить на выполнение условия оптимальности для каждой назанятой клетки.

Таблица 7.6

Матрица планирования		Потребители					Запасы
		B_1	B_2	B_3	B_4	B_5	
Поставщики	V_j / U_i	3	8	12	~~13~~ 7	13	
A_1	+ ~~12~~ −6	10	7	4 $\boxed{2}$	− 1 100	+ 4 $\boxed{3}$	100
A_2	− 11	2 200	− 7 50	10 $\boxed{1}$	+ 6 0	11 $\boxed{1}$	250
A_3	− 11	8	5	3	2	2 200	200
A_4	0	11	+ 8 150	12 100	16	13 50 −	300
Потребности		200	200	100	100	250	850

Если полученный план снова окажется неоптимальным, то следует выполнить вычисления, приведенные в п. 4. Процесс повторяют до тех пор, пока все незанятые клетки не будут удовлетворять условию (7.6). Все вычисления оптимального плана выполнять в одной таблице, поэтому значения x_{ij}, U_i, V_j, а также знаки $\vee$, «–» и «+» следует записывать карандашом и затем стирать.

Построение новой системы потенциалов и проверка всех незанятых клеток на оптимальность требует значительных затрат времени, поэтому рассмотрим порядок изменения системы потенциалов , позволяющих значительно сократить обьем вычислительных работ.

6.Изменение системы потенциалов. В новом опорном плане, записанном в таблице 7.6. пока указаны старые потенциалы. Клетка A_2B_4 прежде свободная, теперь стала занятой. Для занятой клетки должно выполняться условие $U_i + V_j = C_{ij}$. В действительности же $U_2 + V_4 = -1 + 13 = 12 \neq 6$. Следовательно, необходимо либо U_2, либо V_4 уменьшить на шесть. Очевидно, что уменьшать следует тот потенциал, уменьшение которого приводит к наименьшему изменению остальных потенциалов. Так, если изменить V_4, а потенциал U_2 оставить без изменения, то изменеию подлежит только потенциал U_1 ; в противном случае изменяются все остальные потенциалы. Отмечаем потенциал U_2

знаком «!», а потенциал V_4 – знаком «–». Изменение потенциала V_4, так как клетка A_1B_4 занята, потребует изменение потенциала U_1 (отмечаем его знаком «+»); на этом цепочка изменений обрывается.

Таким образом, значения потенциалов, отмеченных знаком «–», уменьшаются, а знаком «+» – увеличиваются (в данном случае на 6 единиц). В резкльтате для всех занятых клеток выполняется соотношение (7.5). Значение потенциала уменьшилось на 6 единиц, поэтому свободные клетки в столбце B_4, не удовлетворяющие условию оптимальности, появиться не могут. Если же такие клетки были, то они могут исчезнуть, так как разность, равная шести , максимальная для всех клеток, в которых нарушено это условия. Свободные клетки, неудовлетворяющие условию оптимальности, могут появиться только в строке (в столбце), потенциал который увеличился. Потенциал U_1 уыеличился на 6 единиц, поэтому незанятые клетки строки A_1 следует проверить на оптимальность: -3<10; 2<7; 6>4; 7>4 . Клетки A_1B_3 и A_1B_5 не довлетворяют этому условию; находим для них величины разностей $u_i + v_j - c_{ij}$ и записываем их в левый нижний угол соответствующих клеток.

Определяем $\max\{2;3;1;1\} = 3$. Клетка A_1B_5 подлежить к загрузке, отмечая ее знаком «+» и установливаем цикл перераспределения, показанный в таблице 7.6 пунктирной линией. Отмечаем вершины цикла попеременно знаками «–» и «+» и находим $\theta_0 = \min\{50;50;100\} = 50$. По циклу перераспределяем 50 единиц груза в клетку A_1B_5 и получаем опорный план, приведенный в таблице 7.7.

Так как значение $\theta_0 = 50$ достигается для двух клеток цикла

Таблица 7.7

Матрица планирования		Потребители					Запасы
		B_1	B_2	B_3	B_4	B_5	
Постав-щики	U_i \ V_j	3	8	12	7	~~15~~ 10	
A_1	– 6	10	7	+ 4 2	– 1 50	4 50	100
A_2	– 1	2 200	– 7 0	10 1	6 50 +	11	250
A_3	+ ~~-11~~ – 8	8	5	3	2	2 200	200
A_4	0	11	+ 8 200	12 100 –	16	13	300
Потребности		200	200	100	100	250	850

отмеченных знаком «–», то полученном опорном плане в клетке A_1B_2

оставлена нулевая перевозка. отмеченных знаком «–», то полученном опорном плане в клетке A_1B_2 оставлена нулевая перевозка.

Заметим, что план в результате последней итерации улучшился на 150 единиц стоймости. Это улучшение находится как произведение количества единиц груза, перемещенного в на свободную клетку, на величину разности в этой клетке. Величина разности $(U_i + V_j) - C_{ij} > 0$ в незанятой клетке показывает, на сколько уменьшится стоимость плана перевозок, если единиц груза перераспределить в эту клетку.

В полученном опорном плане изменяем систему потенциалов и проверяем его на оптимальнсоть. Условию оптимальности не удрвлетворяют две клетки с разностями, равными двум единице, следовательно , грпуз надо перераспределить в клетку A_1B_3. Отмечаем ее знаком «+» и сторим цикл перераспределения, который показан в таблице 7.7 пунктирной линией. Циклы могут имет различную конфигурацию, быть даже самопересекающимися (таблица 7.7).

Отмечаем вершины цикла знаками «–» или «+» и находим величину перераспределения: $\theta_0 = \min\{50; 0; 100\} = 0$. Нулевую перевозку перемещаем в клетку A_1B_3, получаем новый опорный план и вносим изменения в систему потенциалов. Построенная система потенциалов позволяет сделать вывод, что план , приведенный в таблице 7.8, является оптимальным . Его стоимость равна 4150 единиц стоимости.

7.7. Открытая модель транспортной задачи

Транспортные задачи, в которой суммарные запасы и потребности совподают т.е. выполняется условия $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, называется закрытой моделью; в противном случае – открытой. Для открытой модели может быть два случая: а) суммарные запасы превышают суммарные потребности

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j ;$$

б) суммарные потребности превышают суммарные запасы

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j .$$

Линейная функция одинакова, в обоих случаях, изменяется только вид системы ограничиий.

Открытая модель решается приведением к закрытой модели.

Найти минимальное значение линейной функции $Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}x_{ij}$ при ограничениях

$$\begin{cases} \sum_{j=1}^{n} x_{ij} \le a_i, \ i = \overline{1,m} \\ \sum_{i=1}^{m} x_{ij} = b_j, \ j = \overline{1,n} \end{cases} \qquad \text{(случай а)}$$

$$\begin{cases} \sum_{j=1}^{n} x_{ij} \le a_i, \ i = \overline{1,m} \\ \sum_{i=1}^{m} x_{ij} = b_j, \ j = \overline{1,n} \end{cases} \qquad \text{(случай б)}$$

$$x_{ij} \ge 0 \quad (i = 1,2,...,m; \ j = 1,2,...,n)$$

В случае (а), когда суммарные запасы превышают суммарные потребности, вводится фиктивной потребитель B_{n+1}, потребности которого $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$. В случае (б), когда суммарные потребности превышают суммарные запасы, вводится фиктивной поставшик A_{m+1}, запасы которого $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$.

Стоимомть перевозки единицы груза до фиктивного потребителя, тка и стоимость перевозки единицы груза от фиктивногоь поставшика полагают равным нулю, так как груз в обоих случаях не перевозится.

После преобразований задача принимает вид закрытой модели и решается обычным способом. При равных стоимостях перевозки единицы груза от поставшиков к фиктивному потребителю затараты на перевозку груза реальным потребителям минимальны, а фиктивному потребителю будет направлен груз от наименее выгодных поставшиков.
То же самое получаем и в отношении фиктивногогпоставшика.

Прежде чем решать какую-нибьудь транспортную задачу, необходимо сначала проверить, у какой модели она принадлежит, и только после этого составить таблицк ее решения.

Пример. 7.1. Составить план перевозок грузов с наименьшей стоимостью от четырех поставшиков A_i ($i = \overline{1,4}$) соответственно в количествах 100, 400, 100 и 100 единиц к пяти потребителям B_j ($j = \overline{1,5}$) соответственно в количествах 50, 100, 150, 200 и 250 единиц; стоимость перевозок единиц груза приведены в таблице 7.8.

Р е ш е н и е. Вычисляем суммарные запасы и потребности: $\sum_{i=1}^{4} a_i = 700;$

$\sum_{j=1}^{5} b_j = 750$. Потребности превышают запасы на 50 единиц. Необходимо ввести фиктивного поставшика A_{m+1}, обьем запасов которого $a_{m+1} = 50$ единиц. В таблице 7.9 получен оптимальный план. При составлении первоначального опорного плана методом минитмальной стоимости или

Таблица 7.8.

A_i \ B_j	B_1	B_2	B_3	B_4	B_5
A_1	1	6	8	12	16
A_2	16	10	8	6	15
A_3	4	1	9	11	13
A_4	3	2	7	7	15

методом двойного предпочтения необходимо наименьшую стоимость выбирать только среди стоимостей реальных поставшиков и потребителей, а запасы фиктивного поставшика распределять в последнуюю очередь. Это

Таблица 7.9.

Матрица планирования		Потребители					Запасы
		B_1	B_2	B_3	B_4	B_5	
Поставщики	U_i \ V_j	3	5	10	8	17	
A_1	−2	1 50	6	8 50	12	16	100
A_2	−2	16	10	+ 8	6 200	− 15 200	400
A_3	−4	4	− 1 100	9	11	13 0 +	100
A_4	−3	3	+ 2 0	7 100	7	15	100
A_{m+1}	−17	0	0	0	0	0 50	50
Потребности		50	100	150	200	250	750

позволит получить план более близкий к оптимальному. Отмеченное выше используют также при введении фиктивно занятых клеток.

Анализируя оптимальный план задачи, можно сделать следующие выводы. Потребитель B_5 получает 50 единиц груза от фиктивного поставшик, следовательно, его потребности будут не удовлетворены на это же количество единиц. Оптимальный план является не единственным , так как для клетки A_2B_3 сумма потенциалов равна стоимости перевозки $U_2 + V_3 = C_{23}$ и в нее по циклу, показанному в таблице 7.9, можно переместить 100 единиц груза.

В результате перераспределения получен новый опорный план, которой также является оптимальным, так как система по тенциалов останется без изменения и общая стоимость плана перевозок не уменьшается, поскольку $(U_2 + V_3) - C_{23} = 0.$

Контрольные вопросы по главе 7

1. *Постановка транспортной задачи*
2. *Особенности транспортной задачи ограничениями на пропускную способность*
3. *Методы построения начального опорного решения*
4. *Закрытая модель*
5. *Открытая модель*
6. *Метод северо-западного угла*
7. *Метод минимальной стоимости*
8. *Метод двойного предпочтения*
9. *Метод потенциалов*
10. *Невырожденный опорный план.*
11. *Построение системы потенциалов*
12. *Проверка выполнения условия оптимальности для незанятых клеток.*
13. *Выбор клетки, в которую необходимо послать перевозку.*
14. *Построения цикла и определение величины перераспределения груза.*
15. *Изменение системы потенциалов.*

Список исползованной литературы